电子信息与电气学科规划教材·电子信息科学与工程类专业

数字语音处理

及MATLAB仿真

张雪英　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY　http://www.phei.com.cn

电子信息与电气学科规划教材·电子信息科学与工程类专业

数字语音处理及MATLAB仿真

张雪英　编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书系统地阐述了语音信号处理的原理、方法、技术和应用，同时给出了部分内容对应的 MATLAB 仿真源程序。全书共 12 章，第 1 章至第 7 章是基本理论部分，包括语音信号的数字模型、语音信号的短时时域分析和频域分析、语音信号的同态处理、语音信号线性预测分析和矢量量化；第 8 章至第 12 章是应用部分，包括语音编码、语音合成、语音识别、语音增强和语音处理的实时实现。本书内容全面，重点突出，原理阐述深入浅出，注重理论与实际应用的结合，可读性强。

本书可作为高等院校通信工程、电子信息工程、自动控制、计算机技术与应用等专业高年级本科生相关课程的教材，也可供从事语音信号处理研究的研究生和科研人员参考。

图书在版编目(CIP)数据

数字语音处理及 MATLAB 仿真 / 张雪英编著. —北京：电子工业出版社，2010.7
电子信息与电气学科规划教材·电子信息科学与工程类专业
ISBN 978-7-121-11323-9

Ⅰ. ①数… Ⅱ. ①张… Ⅲ. ①语音数据处理－计算机仿真－软件包，MATLAB－高等学校－教材 Ⅳ. ①TN912.3

中国版本图书馆 CIP 数据核字(2010)第 131775 号

策划编辑：凌　毅
责任编辑：李秦华
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1092　1/16　印张：16　字数：410 千字
印　　次：2010 年 7 月第 1 次印刷
印　　数：4000 册　定价：28.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

语言是人类交换信息最方便、最快捷的一种方式，在高度发达的信息社会中，用数字化的方法进行语音的传送、存储、识别、合成和增强等是整个数字化通信网中最重要、最基本的组成部分之一。随着人类步入信息社会步伐的加快，越来越多的地方需要用到语音信号处理知识。语音信号处理作为一门涉及面很广的交叉学科，已经在越来越多高校中的通信工程、电子信息工程、自动控制、计算机技术与应用等专业开设这门课程，语音信号处理的教材也逐渐增多。但目前已有的教材中，基本上以理论阐述为主，内容较深，更适合硕士、博士研究生层次的学生使用，不太适合本科生。

根据教育部关于加强本科生动手实践能力培养的要求，编写本书的目的就是要让本科生通过本课程的学习，了解这门课程的基本理论，同时学会用 MATLAB 语言来处理实际的语音，提高学习本课程的兴趣，培养解决实际问题的能力。

本书最大的特色就是把理论与实际相结合，其中融入了编著者多年从事语音信号处理的科研成果。在阐述基本理论的同时，辅以 MATLAB 源程序，加上详细注释，并配有程序运行结果图。学生们在课后，可以自己动手用工具软件录制语音，照着编写程序，按自己的意图修改程序，进行语音处理的实践。克服以前学生分别学完 MATLAB、语音信号处理课程后，当要用 MATLAB 具体处理一句实际语音时，却不知从何入手的缺陷。

本书主要以高年级本科生和初次学习语音信号处理知识的研究生为读者对象，注重语音信号处理基础知识及主要应用的描述，同时对本领域的最新成果也有简单介绍。全书共 12 章，第 1 章是绪论；第 2 章是语音信号的数字模型；第 3 章是语音信号的短时时域分析；第 4 章是语音信号短时频域分析；第 5 章是语音信号的同态处理；第 6 章是语音信号线性预测分析；第 7 章是矢量量化；第 8 章是语音编码；第 9 章是语音合成；第 10 章是语音识别；第 11 章是语音增强；第 12 章是语音处理的实时实现。附录部分是本书中出现过的专业名词缩写及中英文对照，供大家学习时参考。本书第 1 章至第 7 章属于基本理论部分，所附的 MATLAB 程序较多，第 8 章至第 12 章是语音信号处理技术的应用，这方面的程序一般都比较长，且有一定难度，所以附带的程序较少，且都是相对简单的。可以说，本书是一本关于语音信号处理的入门实践教材，在学习和掌握本书内容的基础上，再进行本专业更深层次的学习是合适的。

本书前 7 章内容可以用做工科高等院校相关专业 **32～40 学时**课程的教程，后 5 章内容可作为选学内容。

本书提供免费的电子课件、MATLAB 仿真程序，读者可登录华信教育资源网 www.hxedu.com.cn，注册后免费下载。

本书由张雪英教授编著，马建芬副教授和李凤莲博士参编，具体分工是：第 1 章、第 2 章、第 3 章、第 4 章、第 5 章、第 7 章、第 8 章、第 9 章、第 10 章、第 12 章由张雪英编写，第 6 章和附录 A 由李凤莲编写，第 11 章由马建芬编写。在本书编写过程中，特别是 MATLAB 程序的调试过程中，得到了太原理工大学信息工程学院电路与系统专业一些硕士生和博士生的帮助，在此表示衷心感谢。

由于编著者水平有限，书中难免存在错误之处，敬请读者批评指正。

编著者

2010 年 3 月

目　　录

第1章 绪 论

1.1 概 述

语言是人类交换信息最方便、最快捷的一种方式，在高度发达的信息社会中，用数字化的方法进行语音的传送、存储、识别、合成和增强等是整个数字化通信网中最重要、最基本的组成部分之一。数字电话通信、高音质的窄带语音通信系统、语言学习机、声控打字机、自动翻译机、智能机器人、新一代计算机语音智能终端及许多军事上的应用等，都要用到语音信号处理技术，随着集成电路和微电子技术的飞速发展，语音信号处理系统逐步走向实用化。

语音信号处理是一门新兴的边缘学科，它是语音学与数字信号处理两个学科相结合的产物。它和认知科学、心理学、语言学、计算机科学、模式识别和人工智能等学科有着紧密的联系。语音信号处理的发展依赖于这些学科的发展，而语音信号处理技术的进步也会促进这些领域的进步。

语音信号处理的目的是要得到某些语音特征参数以便高效地传输或存储；或者是通过某种处理运算以达到某种用途的要求，例如人工合成语音、辨识出讲话者、识别出讲话的内容等。

随着现代科学和计算机技术的发展，除了人与人之间的自然语言的通信方式之外，人机对话及智能机器等领域也开始使用语言。这些人工语言同样有词汇、语法、语法结构和语义内容等。控制论创始人维纳在1950年就曾指出过："通常，我们把语言仅仅看做人与人之间的通信手段，但是，要使人向机器、机器向人及机器向机器讲话，那也是完全办得到的"。通常认为，语音信息的交换大致上可以分为三大类：

① 人与人之间的语言通信：包括语音压缩与编码、语音增强等。

② 第一类人机语言通信问题，指的是机器讲话、人听话的研究，即语音合成。

③ 第二类人机语言通信问题，指的是人讲话、机器听话的情况，即语音识别和理解。

上述这些应用领域构成了语音信号处理技术的主要研究内容。

1.2 语音信号处理的发展

早在一两千年以前，人们便对语言进行了研究。由于没有适当的仪器设备，长期以来，一直是由耳倾听和用口模仿来进行研究。因此，这种语言研究常被称为"口耳之学"，所以对语音只是停留在定性的描写上。

语音信号处理真正意义上的研究可以追溯到1876年贝尔电话的发明，该技术首次用声电、电声转换技术实现了远距离的语音传输。1939年 Homer Dudley 提出并研制成功的第一个声码器，从此奠定了语音产生模型的基础。这一发明在语音信号处理领域具有划时代的意义。19世纪60年代，亥姆霍兹应用声学方法对元音和歌唱进行了研究，从而奠定了语言的声学基础。20世纪40年代，一种语言声学的专用仪器——语谱图仪问世了。它可以把语音的

时变频谱用语图表示出来，从而得出了“可见语言”。1948 年美国 Haskins 实验室研制成功“语音回放机”，该仪器可以把手工绘制在薄膜片上的语谱图自动转换成语音，并进行语音合成。20 世纪 50 年代对语言产生的声学理论开始有了系统的论述。随着计算机的出现，语音信号处理的研究工作得到了计算机技术的帮助，使得过去受人力、时间限制的大量的语音统计分析工作，得以在电子计算机上进行。在此基础上，语音信号处理不论在基础研究方面，还是在技术应用方面，都取得了突破性的进展。下面分别论述语音信号处理的三个主要分支（语音合成技术、语音编码和语音识别技术）的发展和现状。

1.2.1 语音合成

就语音合成技术而言，最早的合成器是 1835 年由 W. von Kempelen 发明，经 Weston 改进的机械式会讲话的机器。该机器完全模仿人的发音生理过程，分别用风箱、特别设计的哨和软管来模拟肺部的空气动力、模拟口腔。而最早的电子式语音合成器是 1939 年 Homer Dudley 发明的声码器，它不是简单地模拟人的生理过程，而是通过电子线路来实现基于语音产生的源－滤波器理论。

但真正具有实用意义的近代语音合成技术是随着计算机技术和数字信号处理技术的发展而发展起来的，主要是采用计算机产生高清晰度、高自然度的连续语音。在语音合成技术的发展中，早期的研究主要是采用参数合成方法。值得提及的是，1973 年 Holmes 发明的并联共振峰合成器和 1980 年 Klatt 发明的串/并联共振峰合成器，只要精心调整参数，这两个合成器都能合成出比较自然的语音。最具代表性的文语转换系统是美国 DEC 公司 1987 年开发的 DECtalk。但是，由于准确提取共振峰参数比较困难，虽然利用共振峰合成器可以得到许多逼真的合成语音，但是整体合成语音的音质难以达到文语转换（TTS）系统的实用要求。

自 20 世纪 80 年代末期至今，语音合成技术又有了新的进展，特别是 1990 年提出的基音同步叠加（PSOLA）方法，使基于时域波形拼接方法合成的语音的音色和自然度大大提高。20 世纪 90 年代初，基于 PSOLA 技术的法语、德语、英语、日语等语种的文语转换系统都已经研制成功。这些系统的自然度比以前基于 LPC 方法或共振峰合成器的文语合成系统的自然度要高，并且基于 PSOLA 方法的合成器结构简单，易于实时实现，有很大的商用前景。

我国的汉语语音合成研究起步较晚，但从 20 世纪 80 年代初就基本上与国际研究同步发展。大致也经历了共振峰合成、LPC 合成到应用 PSOLA 技术的过程。在国家 863 计划、国家自然科学基金委员会、国家攻关计划、中国科学院有关项目等支持下，汉语文语转换系统研究近年来取得了令人瞩目的进展，其中不乏成功的例子，如 1993 年中国科学院声学所研制的 KX-PSOLA，1995 年研制的联想佳音；清华大学在 1993 年研制的 TH_SPEECH；1995 年中国科技大学研制的 KDTALK 等系统。这些系统基本上都采用基于 PSOLA 方法的时域波形拼接技术，其合成汉语普通话的可懂度、清晰度达到了很高的水平。然而同国外其他语种的文语转换系统一样，这些系统合成的句子及篇章语音机器味较浓，其自然度还不能达到用户可广泛接受的程度，从而制约了这项技术大规模进入市场。

现阶段语音合成的最大进展是已经能够实时地将任意文本转换成连续可懂的自然语句输出。文语转换使得数据通信和语音通信在终端一级实现交融，人们将有望在获取 Internet 信息时，使短消息服务、电子邮件等多数以文本方式提供的信息也能用语音的方式输出。语音合成技术经历了从参数合成到拼接合成，再到两者的逐步结合，其不断发展的动力是人们认知水平和需求的提高。

1.2.2 语音编码

语音编码的目的就是在保证一定语音质量的前提下，尽可能降低编码比特率，以节省频率资源。语音编码技术的研究开始于1939年军事保密通信的需要，贝尔电话实验室的Homer Dudley提出并实现了在低带宽电话电报电缆上传输语音信号的通道声码器，成为语音编码技术的鼻祖。直到20世纪70年代，国际电联(ITU-T，原CCITT)于1972年发布了64kbit/s脉冲编码调制(PCM)语音编码算法的G.711建议，它被广泛应用于数字通信、数字交换机等领域，从而占据统治地位。1980年美国政府公布了一种2.4kbit/s的线性预测编码标准算法LPC-10，这使得在普通电话带宽中传输数字电话成为可能。ITU-T也于20世纪80年代初着手研究低于64kbit/s的非PCM编码算法，并于1984年通过了32kbit/s ADPCM语音编码G.721建议，它不仅可以达到与PCM相同的语音质量，而且具有更优良的抗误码性能。1988年美国又公布了一个4.8kbit/s的码激励线性预测(CELP)编码算法。与此同时，欧洲也推出了一个16kbit/s的规则脉冲激励线性预测(RPE-LPC)编码算法。这些算法的语音质量都能达到较高的水平，大大超过LPC声码器的质量。进入20世纪90年代，随着因特网在全球范围的兴起，人们对能在网络上传输语音的VoIP技术兴趣大增，由此，IP分组语音通信技术获得了突破性进展和实际应用。ITU-T于1992年公布了16kbit/s低延迟码激励线性预测编码(LD-CELP)的G.728建议。它以其较小的延迟、较低的速率、较高的性能在实际中得到广泛的应用，也成为分组化语音通信的可选算法之一。1996年ITU-T发布了码率为5.3/6.4kbit/s的G.723.1标准。在1995年11月ITU-T SG15全会上通过了共轭代数码激励线性预测(CS-ACELP)的8kbit/s语音编码G.729建议，并于1996年6月ITU-T SG15会议上通过G.729附件A：减少复杂度的8kbit/s CS-ACELP语音编解码器，正式成为国际标准。这几种语音编码算法也成为分组化语音通信的可选算法。

语音编码技术主要有两个努力方向：一是中低速率的语音编码的实用化及如何在实用化过程中进一步提高其抗干扰、抗噪声能力；另一个是如何进一步降低其编码速率。目前已能在5～6kbit/s的速率上获得高质量的重建语音，下一个目标则是要在4kbit/s的速率上获得短延时、高质量的重建语音。特别是对中长延时编码，人们正在研究其更低速率(如400～1200bit/s)的编码算法。当编码速率降至2.4kbit/s以下时，CELP算法即使应用更高效的量化技术也无法达到预期的指标，需要其他一些更符合低速率编码要求的算法，目前比较好的算法有正弦变换编码(STC)、混合激励线性预测编码(MELPC)、时频域插值(TFI)编码、基音同步激励线性预测(PSELP)编码等，同时还要求引入新的分析技术，如非线性预测、多精度时频分析技术(包括子波变换技术)、高阶统计分析技术等，这些技术更能挖掘人耳听觉掩蔽等感知机理，更能以类似人耳的特性作为语音的分析与合成，使语音编码系统更接近于人类听觉器官的处理方式工作，从而在低速率语音编码的研究上取得突破。

20世纪90年代中期到现在，第三代移动通信技术逐渐成熟并走向商用，变速率语音编码和宽带语音编码得到了迅速的发展，不断有新的国际标准和地区标准公布。应用于第三代移动通信的变速率语音编码主要有可变速率码激励线性预测(QCELP)、增强型变速率编码器(EVRC)、自适应多速率(AMR)编码器、自适应多速率宽带(AMR-WB)编码器、可选模式声码器(SMV)和变速率多模式宽带(VMR-WB)编码器等。宽带语音的发展也经历了一个过程，1988年国际电联通过了第一个宽带语音编码器标准G.722，基于子带自适应差分脉码调制(SB-ADPCM)编码原理，速率为64kbit/s、56kbit/s和48kbit/s。宽带语音编码器的合成语音

更自然，非常适合应用到电视电话会议中。早期的宽带语音编码器的缺点就是编码效率不高，64kbit/s 的速率不利于在系统中实现。1999 年 ITU-T 公布了新的宽带语音编码国际标准 G.722.1，降低了编码速率(24kbit/s 和 32kbit/s)。2002 年 ITU-T 在对以往宽带语音编码算法改进的基础上提出 G.722.2 标准，由 9 种速率的语音模式组成，编码速率较低，而且可以根据无线环境和本地容量需求动态选择。变速率语音编码理论上仍属于 CELP，但在“变”上有了新的研究，由此引入了相关技术的研究，包括：用来检测语音通信时是否有语音存在的语音激活检测(VAD)技术、为突出“变”字而进行速率判决(RDA)的自适应技术、为避免语音帧丢失后带来负面效应的差错隐藏(ECU)技术、为克服背景噪声不连续的舒适背景噪声生成(CNG)技术等。这些相关技术的应用使变速率语音编码之后的语音合成效果几乎没有降低。随着移动通信的飞速发展，用变速率语音编码来提高频带的有效利用率，将是未来数字蜂窝和微蜂窝网的必然发展趋势。

1.2.3 语音识别

与机器进行语音交流，让机器明白你说什么，这是人们长期以来梦寐以求的事情。而语音识别技术就是让机器通过识别和理解过程把语音信号转变为相应的文本或命令的高技术。由于语音本身所固有的难度，让机器识别语音的困难在某种程度上就像一个外语不好的人听外国人讲话一样，它和不同的说话人、不同的说话速度、不同的说话内容及不同的环境条件有关。语音信号本身的特点造成了语音识别的困难，这些特点包括多变性、动态性、瞬时性和连续性等。根据在不同限制条件下的研究任务，产生了不同的研究领域。这些领域包括：①根据对说话人说话方式的要求，可以分为孤立字语音识别系统、连接字语音识别系统及连续语音识别系统；②根据对说话人的依赖程度可以分为特定人和非特定人语音识别系统；③根据词汇量大小，可以分为小词汇量、中等词汇量、大词汇量及无限词汇量语音识别系统。

语音识别的研究工作真正开始于 20 世纪 50 年代 AT&T 贝尔实验室的 Audry 系统，它是第一个可以识别 10 个英文数字的语音识别系统。1956 年 RAC 实验室的 Olson 等人也独立地研制出 10 个单音节词的识别系统，系统采用从带通滤波器组获得的频谱参数作为语音的特征。1959 年 Fry 和 Denes 等人采用频谱分析和模式匹配来进行识别决策构建音素识别器来辨别 4 个元音和 9 个辅音。同年，MIT 林肯实验室采用声道的时变估计技术研究 10 个元音的识别。

但语音识别的研究真正取得实质性进展，并将其作为一个重要的课题开展则是在 20 世纪 60 年代末。这一方面是因为计算机的计算能力有了迅速的提高，能够提供实现复杂算法的软件、硬件环境；另一方面，数字信号处理理论和算法在当时有了蓬勃发展，从而自 20 世纪 60 年代末开始引起了语音识别的研究热潮。这时期的重要成果是提出了动态规划(DP)和线性预测编码(LPC)分析技术，其中后者较好地解决了语音信号产生模型的问题，对整个语音识别、语音合成、语音分析、语音编码的研究发展产生了深远影响。

20 世纪 70 年代，语音识别领域取得了突破性进展。在理论上，LPC 技术得到进一步发展，动态时间弯折(DTW)技术基本成熟，特别是提出了矢量量化(VQ)和隐马尔可夫模型(HMM)理论。在实践上，首先在孤立词识别方面，由日本学者 Sakoe 给出了使用动态规划方法(DP)进行语音识别的途径——DP 算法。DP 算法是把时间规整和距离测度计算结合起来的一种非线性规整技术，这是语音识别中一种非常成功的匹配算法，并在小词汇量中获得了成功，从而掀起了语音识别的研究热潮。另外，就是学者 Itakura 基于语音编码中广泛使用的

LPC 技术，通过定义基于 LPC 频谱参数的合适的距离测度，成功地将其应用到语音识别中。同时，以 IBM 为首的一些语音研究单位还着手开展了连续语音识别的研究。

在 20 世纪 70 年代末和 80 年代初，Linda、Buzo、Gray 等人解决了矢量量化码本生成的方法，并将矢量量化成功地应用到语音编码中，从此矢量量化技术很快被推广应用到其他领域。

从 20 世纪 80 年代开始，语音识别研究进一步走向深入，就是识别算法从模式匹配技术转向基于统计模型的技术，更多地追求从整体统计的角度来建立最佳的语音识别系统。HMM 技术就是其中的一个典型技术。最早将 HMM 用于语音识别是 20 世纪 70 年代中期，但对 HMM 的全面研究和大规模应用是 20 世纪 80 年代以后的事情。它受到广泛重视的原因是：马尔可夫链可以用来描述蕴藏于观察数据中的时变特性，这使得它能处理语音信号中常常出现的非平稳特性（即时变特性）。它不仅能用于描述各种不同层次的语音单元，甚至可以描述 VQ 中的任一码字或由声学特征定义的任一种声学单元，并且由小单元模型组成大单元模型［音节（或音素）→单词→句子］。由 Viterbi 解码可得到与语音序列相对应的最佳状态序列，从而得到语音单元的最佳分割，使子词单元的使用非常方便，大大避免了训练和识别时的分割困难，使连续语音识别问题得到解决。随着对 HMM 的深入研究和在语音识别中的需要，许多新的算法产生，如估计、平滑、外插、建立时间模型、话者自适应等，使得这一技术在语音识别中有了更深入的应用。到目前为止，HMM 方法仍然是语音识别研究中的主流方法，并使得大词汇量连续语音识别系统的开发成为可能。在 20 世纪 80 年代末，由美国卡内基梅隆大学用 VQ/HMM 实现 997 个词的非特定人连续语音识别系统 SPHINX 成为世界上第一个高性能的非特定人、大词汇量、连续语音识别系统。这些研究开创了语音识别的新时代。

20 世纪 80 年代中期重新开始的人工神经网络（ANN）研究，也给语音识别带来一片新的生机。由于 ANN 具有自组织和自动学习各种复杂分类边界的能力，以及很强的区分能力，使它特别适用于语音识别这一特殊的分类问题。人们将 ANN 和 HMM 在同一语音识别系统中结合使用，即由 ANN 完成静态的模式分类问题，而用 HMM 甚至传统的 DP 来完成时间对准问题。从实验结果来看，这种思想可行而且有效，并能使 ANN 比较容易地用于连续语音识别问题。语音识别常用的 ANN 有：时间延迟神经网络 TDNN、递归神经网络 RNN、自组织神经网络 SONN、学习矢量量化 LVQ 及混合语音识别系统。

进入 20 世纪 90 年代，随着多媒体时代的来临，迫切要求语音识别系统从实验室走向实用。许多发达国家如美国、日本以及 IBM、Apple、AT&T、NTT 等著名公司都为语音识别系统的实用化开发研究投以巨资。在 20 世纪 90 年代初期，开始出现孤立语音的英文听写机系统，在 1997 年开始出现基于说话人自适应的连续语音听写系统，并达到一定的实用化程度。从语音识别的进展来看，国际上孤立词识别系统已经扩大到数万个，特定说话人或非特定说话人的连续语音识别系统已达到了很高的识别率。从研究领域来看，在连续语音中识别关键词的研究以及多种语言之间的自动翻译、语音检索等已成为比较热门的课题。随着网络技术和语音研究工作的迅速发展，出现了语种识别技术、基于语音的情感技术、嵌入式语音识别技术等一些新的研究方向。

在国内，语音识别的研究工作起步于 20 世纪 50 年代，但是除中科院声学所外，大多数单位是 20 世纪 70 年代末及 80 年代初才开始的。到 20 世纪 80 年代末，以汉语全音节识别为主攻方向的研究已经取得相当大的进展，一些汉语输入系统已向实用化迈进。20 世纪 90 年代初，在国家“863 计划”支持下，国家 863 智能计算机专家组为语音识别技术研究专门立项。清华大学与中科院自动化所等单位在汉语听写机原理样机的研制方面开展了卓有成效的研究。

北京大学在说话人识别方面也做了很好的研究。近些年，在我国科研人员长期艰苦努力下，我国在语音技术研究水平和原型系统开发方面达到了世界级的水平，做出了当之无愧的成果。在中国科学院自动化研究所模式识别国家重点实验室，汉语非特定人、连续语音听写机系统的普通话系统，其错误率可以控制在10%以内的的水平，并具有非常好的自适应功能。尤其是在国内外首创研究开发了汉语自然口语的人机对话系统和汉语到日语、英语的直接语音翻译系统，为在未来发展民族化的语音产业打下了非常坚实的技术基础。清华大学王作英教授提出的非齐次基于段长分布的隐马尔可夫模型(DDBHMM)可以说是对语音识别模型算法的一次重大革新。以此理论为指导所设计的语音识别听写机系统在1994-1998年的全国语音识别系统评测中取得三连冠，从而显示了这一新模型的生命力和在这一研究领域内的领先水平。目前，我国语音识别技术的研究已取得令人瞩目的成绩，其基础研究涉及汉语语音学、听觉模型、人工神经网络、小波变换、分形维数和支持向量机等理论，其研究成果必将推动我国语音识别技术研究迈上新台阶。

1.3 语音信号处理的应用及新方向

语音信号处理技术是计算机智能接口与人机交互的重要手段之一。从目前和整个信息社会发展趋势看，语音技术有很多的应用。语音技术包括语音识别、说话人的鉴别和确认、语种的鉴别和确认、关键词检测和确认、语音合成、语音编码等，但其中最具有挑战性和最富有应用前景的为语音识别技术。

首先对于说话人识别技术，近年来已经在安全加密、银行信息电话查询服务等方面得到了很好的应用。此外，说话人识别技术也在公安机关破案和法庭取证方面发挥着重要的作用。其次对于语音识别技术而言，在一些应用领域中正成为一个关键的具有竞争力的技术。例如，在声控应用中，计算机可识别输入的语音内容，并根据内容来执行相应的动作，这包括了声控电话转换、声控语音拨号系统、声控智能玩具、信息网络查询、家庭服务、宾馆服务、旅行社服务系统、医疗服务、股票查询服务和工业控制等。在电话与通信系统中，智能语音接口正在把电话机从一个单纯的服务工具变成为一个服务的“提供者”和生活“伙伴”；使用电话与通信网络，人们可以通过语音命令方便地从远端的数据库系统中查询与提取有关的信息；随着计算机的小型化，键盘已经成为移动平台的一个很大障碍，想象一下如果手机仅仅只有一个手表那么大，再用键盘进行拨号操作已经是不可能的。再者，语音信号处理还可用于自动口语分析，如声控打字机等。随着计算机和大规模集成电路技术的发展，这些复杂的语音识别系统也已经完全可以制成专用芯片，大量生产。在西方经济发达国家，大量的语音识别产品已经进入市场和服务领域。一些用户交换机、电话机、手机已经包含了语音识别拨号功能，还有语音记事本、语音智能玩具等产品也包含了语音识别与语音合成功能。人们可以通过电话网络用语音识别口语对话系统查询有关的机票、旅游、银行信息，并且取得很好的结果。

就语音合成而言，它已经在许多方面得到了实际的应用并发挥了很大的社会作用。例如，公交汽车上的自动报站、各种场合的自动报时、自动报警、手机查询服务和各种文本校对中的语音提示等。在电信声讯服务中的智能电话查询系统中，采用语音合成技术可以弥补以往通过电话进行静态查询的不足，满足海量数据和动态查询的需求，如股票、售后服务、车站查询等信息；也可用于基于微型机的办公、教学、娱乐等智能多媒体软件，例如语言学习、教学软件、语音玩具、语音书籍等；也可与语音合成技术与机器翻译技术结合，实现语音翻译等。

对于语音编码而言，随着人类社会信息化进程的加快，语音编码技术也正在迅速发展，在移动通信、卫星通信、军事保密通信、信息高速公路和 IP 电话通信中得到了广泛的应用。例如低速率语音编码技术解决了信道容量问题。光纤通信技术使有线通信的信道容量得到了缓解，但对于信道价格昂贵的卫星通信及线路铺设艰难的边远山区通信，仍希望能在现有信道上得到更大的通信容量。再者由于数字加密技术具有高度可靠性，一般在军事保密通信中采用低速率语音编码器，以便对经过压缩编码后的语音数据进行加密处理，然后在窄带信道上进行传输。个人移动通信、语音存储、多媒体通信、数字数据网(DDN)中也用到语音通信技术。目前语音编码的算法发展较快，它可应用的范围也相当广泛，除了上述应用外，未来的 ISDN、卫星通信、移动通信、微波接力通信和信息高速公路以及保密电话等无一例外地都会采用低速率语音编码技术。

随着信息技术的不断发展，尤其是网络技术的日益普及和完善，语音信号处理技术正发挥着越来越重要的作用，并且出现了一些新的方向。

① 基于语音的信息检索。随着网络技术及数字图书馆技术的发展，针对于传统的基于文本信息的检索技术，基于语音识别的信息检索技术正成为当今的研究热点。

② 基于语音识别的广播新闻的自动文摘技术的研究。由于广播、电视中的发音较为标准规范，在识别中避免了说话人发音上的不规范，有利于语音识别系统性能的提高。

③ VoIP 技术。它是通过 TCP/IP 网络，而不是传统的电话网络来传输语音的新的通信方式，通常称为 IP 电话技术。它是在网络上对压缩的语音数据以数据包的形式进行传输和识别。随着手机、PDA 等移动电子设备的发展，嵌入式语音识别算法的研究已逐渐成为研究的热点。

④ 语音训练与校正技术也是近年来语音信号处理的一个重要方向。现在越来越多的人希望掌握其他非母语语言，以便方便地进行交流。因此语言学习机已成为当今外语学习者的有利工具。

⑤ 语种识别。语种识别是近年来新出现的研究方向，它是通过分析处理一个语音片断来判别其所属语音的种类，本质上属于语音识别的研究范畴。

⑥ 基于语音的情感处理研究。在人与人的交流中，除了语音信息外，非语言信息也起着重要的作用。为了使人机交流更自然、更人性化，基于语音的情感处理研究也是非常必要的。

1.4 语音信号处理过程的总体结构

信息加工和处理的一般流程如图 1.1 所示。

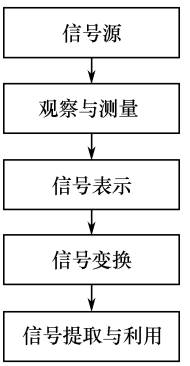


图 1.1 信号加工和处理的一般流程

在语音信号的具体情况下，信息源就是说话的人，通过观察和测量得到的就是语音的波形。信号处理包括以下几个内容，首先根据一个给定的模型得到这一信号的表示；然后再用某种高级的变换把这一信号变成一种更加方便的形式；最后一步是信息的提取和使用，这一步可由听者来完成，也可由机器自动完成。

所以，语音信号处理一般有两个任务：第一，它是一种工具，利用它可以得到语音信号的一般表示，这种表示可以用波形表示也可用参数形式表示；第二，把信号从一种形式变换到另一种形式，变换后的表示形式虽然从性质上讲它的普遍性可能小一些，但对某一特殊应用却是更加合适。由此从总体上来看，语音信号处理过程可以用统一的框架来表示，其基本的结构框图如图 1.2 所示。

从图 1.2 可以看出：无论是语音识别还是语音编码与合成，对于输入的语音信号首先要进行预处理，对信号进行适当的放大和增益控制，并进行反混叠滤波来消除工频信号的干扰；然后进行数字化，将模拟信号转换为便于计算机处理的数字信号；随后对数字语音信号进行分析，提取一定的反映语音信息的参数；最后根据语音信号处理任务的不同，采用不同的处理方法。语音识别技术分为两个阶段：语音识别和训练阶段。在训练阶段，对用特定的参数形式表示的语音信号进行相应的处理，获得表示识别基本单元共性特点的标准数据，以此构成参考模板，并将所有能识别的基本单元的参考模板结合在一起，形成参考模式库；在识别阶段，将待识别的语音信号经特征提取后逐一与参考模型库中的各个模板按某种原则进行比较，找出最相似的参考模板所对应的发音，即为识别结果。对于语音编码技术来说，为了对语音信号进行有效的传输，需要对语音信号以某种算法进行编码，并在接受端进行解压缩。对于语音信号的合成，则是对编码后的信号进行储存。

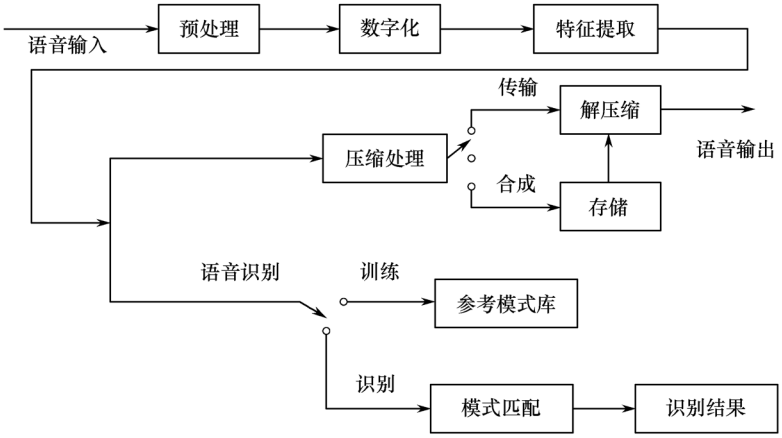


图 1.2　语音处理过程的结构框图

1.5　MATLAB 在数字语音信号处理中的应用

数字语音信号处理是将数字信号处理与语音学相结合，解决现代通信领域中人与人、人与机器之间的信息交流的学科。近几年来语音信号处理学科在世界范围内已取得了飞速的发展，又因为 MATLAB 是一种功能强大、效率高、交互性好的数值计算和可视化计算机高级语言，它将数值分析、信号处理和图形显示有机地融合一体，形成了一个极其方便、用户界面友好的操作环境。随着 MATLAB 的不断发展，其功能越来越强大，广泛应用于数字语音信号处理、数值图像处理、仿真、自动控制、小波分析和神经网络等领域。同时又由于 MATLAB 具有大量的信号处理工具箱并能利用非线性动态系统分析工具 Simulink 等优点，所以近年来 MATLAB 已成为数字信号处理的有利工具，因此也成为学习语音信号处理和进行研究工作的仿真软件工具。

下面简要介绍 MATLAB 在数字语音信号中的几方面应用。

① 通过 MATLAB 可以对数字化的语音信号进行时频域分析。通过 MATLAB 可以方便地展现语音信号的时域及频域曲线，并且根据语音的特性对语音进行分析。例如，清浊音的幅度差别、语音信号的端点、信号在频域中的共振峰频率、加不同窗和不同窗长对信号的影响、

LPC 分析、频谱分析等。

② 通过 MATLAB 可以对数字化的语音信号进行估计和判别。例如，根据语音信号的短时参数，以及不同语音信号的短时参数的性质对一段给定的信号进行有无声和清浊音的判断、对语音信号的基音周期进行估计等。

③ 通过利用 MATLAB 编程对语音信号进行处理。由于 MATLAB 是一种面向科学和工程计算的高级语言，允许用数学形式的语言编程，又有大量的库函数，所以编程简单、编程效率高、易学易懂。我们可以对信号进行加噪和去噪、滤波、截取语音等，也可进行语音编码、语音识别、语音合成的编程等。

本书中的程序实例均用 MATLAB 语言编写，供大家上机实践时参考。

第 2 章　语音信号的数字模型

2.1　概　　述

为了用数字信号处理方法对语音信号进行处理，首先需要建立语音信号产生的数字模型，因此，我们必须在对人的发声器官和发声机理进行研究的基础上，才能建立精确的模型。但是，由于人类语音产生过程的复杂性和语音信息的丰富性及多样性，迄今为止还没有找到一种能够精确描述语音产生过程和所有特征的理想模型。本章介绍的线性模型是一种经典的模拟语音信号产生过程比较成功的模型，它简单实用，是学习语音信号处理理论的基础。

作为接受语音信息的人耳听觉系统，其听觉机理也是很复杂的。听觉模型的精确建立对于语音识别和理解是非常重要的，但是，目前人们对听觉机理的了解比对发音机理的了解少得多。本章重点介绍语音信号产生的数字模型，对语音信号的特性和听觉特性做一般介绍。

2.2　语音的发声机理

2.2.1　人的发声器官

人类的语音是由人的发声器官在大脑控制下的生理运动产生的。人的发声器官由 3 部分组成：①肺和气管产生气源；②喉和声带组成声门；③由咽腔、口腔、鼻腔组成声道，参见图 2.1 所示的发声器官机理模型。

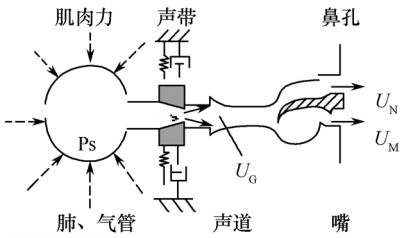


图 2.1　发声器官机理模型

肺的发声功能主要是产生压缩气体，通过气管传送到声音生成系统。气管连接着肺和喉，它是肺与声道联系的通道。

喉是控制声带运动的软骨和肌肉的复杂系统，它主要包括：环状软骨、甲状软骨、杓状软骨和声带。其中声带是重要的发声器官，它是伸展在喉前、后端之间的褶肉，如图 2.2 所示，前端由甲状软骨支撑，后端由杓状软骨支撑，而杓状软骨又与环状软骨较高部分相联。这些软骨在环状软骨上的肌肉的控制下，能将两片声带合拢或分离。声带之间的间隙称为声门。声带的声学功能主要是产生激励。位于喉前端呈圆形的甲状软骨称为喉结。

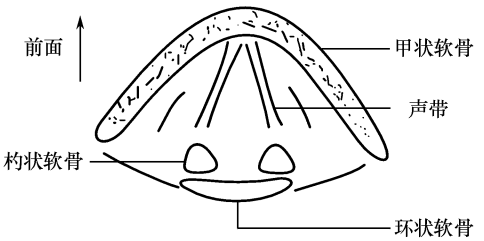


图 2.2　喉的平面解剖示意图

声道是指声门至嘴唇的所有发音器官，其纵剖面图如图 2.3 所示。其中包括：咽喉、口腔和鼻腔。口腔包括上下唇、上下齿、上下齿龈、上下腭、舌和小舌等部分。上腭又分为硬腭和软腭两

部分；舌又分为舌尖、舌面和舌根三部分。鼻腔在口腔上面，靠软腭和小舌将其与口腔隔开。当小舌下垂时，鼻腔和口腔便耦合起来，当小舌上抬时，口腔与鼻腔是不相通的。口腔和鼻腔都是发声时的共鸣器。口腔中各器官能够协同动作，使空气流通过时形成各种不同情况的阻碍并产生振动，从而发出不同的声音来。声道可以看成是一根从声门一直延伸到嘴唇的具有非均匀截面的声管，其截面积主要取决于唇、舌、腭和小舌的形状和位置，最小截面积可以为零（对应于完全闭合的部位），最大截面积可以达到约 $20cm^2$。在产生语音的过程中，声道的非均匀截面又是随着时间在不断地变化的。成年男性的声道的平均长度约为 17cm。当小舌下垂使鼻腔和口腔耦合时，将产生出鼻音来。

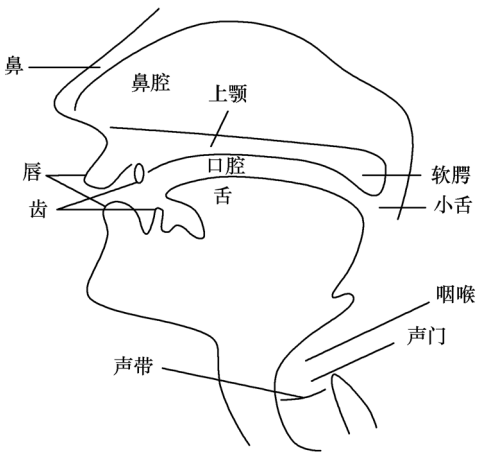


图 2.3　声道纵剖面图

2.2.2　语音生成

图 2.1 为语音生成机理模型。空气由肺部排入喉部，经过声带进入声道，最后由嘴辐射出声波，这就形成了语音。在声门（声带）以左，称为“声门子系统”，它负责产生激励振动；右边是“声道系统”和“辐射系统”。当发不同性质的语音时，激励和声道的情况是不同的，它们对应的模型也是不同的。

1. 发浊音的情况

空气流经过声带时，如果声带是崩紧的，则声带将产生张弛振动，即声带将周期性地启开和闭合。声带启开时，空气流从声门喷射出来，形成一个脉冲，声带闭合时相应于脉冲序列的间隙期。因此，这种情况下在声门处产生出一个准周期脉冲状的空气流。该空气流经过声道后最终从嘴唇辐射出声波，这便是浊音语音。这个准周期脉冲的周期即为基音周期。声门处产生的准周期脉冲其周期、宽度以及形状与声带的长度、厚度及张力等参数有关。声带越短、厚度越薄、张力越大，则听起来感觉的音调就越高，也就是浊音的基音频率越高。因此，基音频率是由声带张开闭合的周期所决定的。男性的基音频率一般为 50～250Hz，女性基音频率为 100～500Hz。

2. 发清音的情况

空气流经过声带时，如果声带是完全舒展开来的，则肺部发出的空气流将不受影响地通过声门。空气流通过声门后，会遇到两种不同情况。一种情况是，如果声道的某个部位发生收缩形成了一个狭窄的通道，当空气流到达此处时被迫以高速冲过收缩区，并在附近产生出空气湍流，这种湍流空气通过声道后便形成所谓摩擦音或清音。另一种情况是，如果声道的某个部位完全闭合在一起，当空气流到达时便在此处建立起空气压力，闭合点突然开启便会让气压快速释放，经过声道后便形成所谓爆破音。这两种情况下发出的音称为清音。

当声音产生后，便沿着声道进行传播。声道可以看成一根具有非均匀截面的声管，在发声时起着共鸣器的作用。声音进入声道后，其频谱必定会受到声道的共振特性的影响，声道具有一组共振频率，称为共振峰频率或共振峰。声道的频谱特性便主要地反映出这些共振峰的不同位置以及各个峰的频带宽度。共振峰及其带宽取决于声道的形状和尺寸，因而不同的语音对应于一组不同的共振峰参数。

2.3 语音的听觉机理

听觉是接受声音并将其转换成神经脉冲的过程。大脑受到听觉神经脉冲的刺激感知为确定的含义是一个非常复杂的过程，至今尚不完全清楚。

2.3.1 听觉器官

人的听觉器官分为3个部分：外耳、中耳和内耳，如图2.4所示。

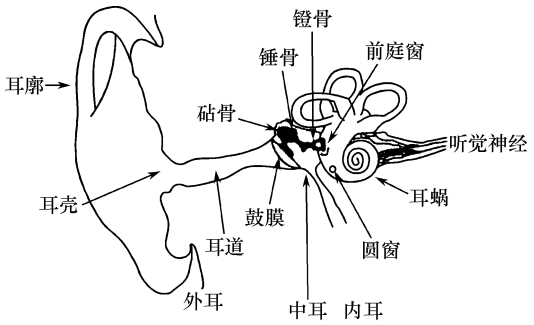


图2.4 人耳结构示意图

外耳由位于头颅两侧呈贝壳状和向内呈S状弯曲的外耳道组成，主要包括：耳廓、耳壳和外耳道组成，它的主要作用是收集声音、辨别声源，并对某些频率的声音有扩大作用。声音沿外耳道传送至鼓膜，外耳道有许多共振频率，恰好落在语音频率范围内。

中耳主要由鼓膜和听骨链组成。听骨链由三块听小骨组成，分别称为锤骨、砧骨和镫骨。其中锤骨柄与鼓膜相连，镫骨底板与耳蜗的前庭窗相连。声音经鼓膜至内耳的传输过程主要由听骨链来完成。由于鼓膜的面积比前庭窗大出许多倍(55∶3.2)，听骨链有类似杠杆的作用，所以人的声音从鼓膜到达内耳时，能量扩大了20多倍，补充了声音在传播过程中的能量消耗。

由于中耳将气体运动高效地转为液体运动，所以它实际上起到一种声阻抗匹配的作用，由此可以看出，整个中耳的主要生理功能是传音，即将声音由外耳道高效地传入耳蜗。

从上述分析可以看出，中耳的主要功能是改变增益，还有就是对外耳和内耳进行匹配阻抗。

内耳是颅骨腔内的一个小而复杂的体系，由前庭窗、圆窗和耳蜗构成，前庭窗在听觉机制中不起什么作用，圆窗可以为不可压缩液体缓解压力，耳蜗是内耳的主要器官，它是听觉的受纳器，形似蜗牛壳，为螺旋样骨管。蜗底面向内耳道，耳蜗神经穿过此处许多小孔进入耳蜗。耳蜗中央有呈圆锥形骨质的蜗轴，从蜗轴有螺旋板伸入耳蜗管内，由耳蜗底盘旋上升，直到蜗顶。它由三个分隔的部分组成：鼓阶、中阶和前庭阶。鼓阶与中耳通过圆窗相连，前庭阶与中耳的镫骨由前庭窗的膜相连，鼓阶和前庭阶在耳蜗的顶端即蜗孔处是相通的。中阶的底膜称为基底膜(Basilar membrane)，在基底膜之上是科蒂氏器官(Organ of Corti)，它由耳蜗覆膜(Tectorial membrane)、外毛细胞(Outer hair cell)及内毛细胞(Inner hair cell)构成。图2.5给出了耳蜗未展开时的内耳。

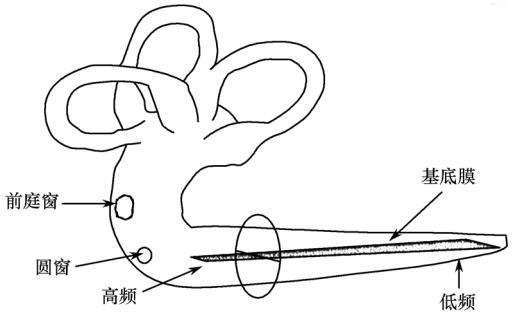


图2.5 耳蜗未展开时的内耳

2.3.2 耳蜗的信号处理机制

当声音经外耳传入中耳时，镫骨的运动引起耳蜗内流体压强的变化，从而引起行波沿基底膜的传播。图 2.6 是流体波的简单表示。在耳蜗的底部基底膜的硬度很高，流体波传播得很快。随着波的传播，膜的硬度变得越来越小，波的传播也逐渐变缓。不同频率的声音产生不同的行波，而峰值出现在基底膜的不同位置上。频率较低时，基底膜振动的幅度峰值出现在基底膜的顶部附近；相反，频率较高时，基底膜振动的幅度峰值出现在基底膜的基部附近（靠近镫骨）。如果信号是一个多频率信号，则产生的行波将沿着基底膜在不同的位置产生最大的幅度如图 2.7 所示。从这个意义上讲，耳蜗就像一个频谱分析仪，将复杂的信号分解成各种频率分量。

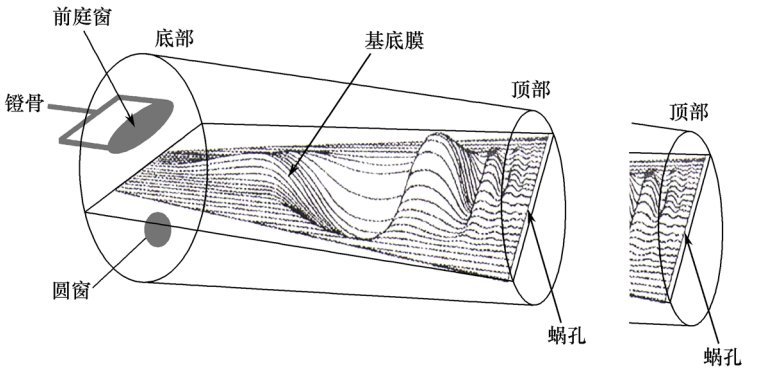


图 2.6　耳蜗内流体波的简单表示

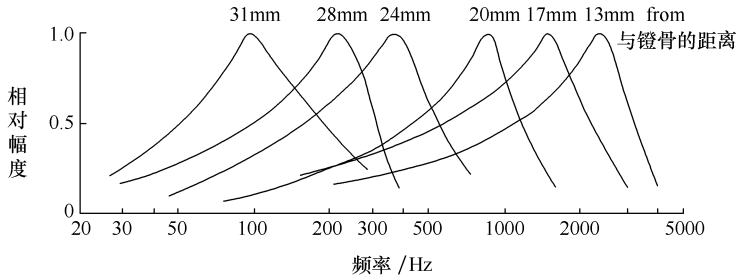


图 2.7　基底膜上 6 个不同点的频率响应

基底膜的振动引起毛细胞的运动，使得毛细胞上的绒毛发生弯曲。绒毛向一个方向的弯曲会使细胞产生去极化，即开启离子通道产生向内的离子流，从而使传入神经开放增加。而绒毛向另一个方向弯曲时，则会引起毛细胞的超极化，即增加细胞膜电位，从而导致抑制效应。因此，内毛细胞对于流体运动速度而言，就像一个自动回零的半波整流器。在基底膜不同部位的毛细胞具有不同的电学与力学特征。在耳蜗的基部，基底膜宽而柔和，毛细胞及其绒毛也较长而柔和。正是由于这种结构上的差异，因此它们具有不同的机械谐振特性和电谐振特性。有学者认为这种差异可能是确定频率选择性的最重要的因素。外毛细胞可在中枢神经系统的控制下调节科尔蒂器官的力学特性，内毛细胞则负责声音检测并激励传入神经发放，而内外毛细胞通过将其绒毛插入共同的耳蜗覆膜而耦合。这样，外毛细胞性质的变化可以调节内毛细胞的调谐，使整个耳蜗的动态功能处于大脑控制之下。

对于听神经如何表达声音信息，目前有两种流行的解释，一种是“发放率－位置表达”，另一种则是“时间－位置表达”，即听神经纤维与刺激同步发放。但这两种解释尚不能完满地解释对不同复杂声音刺激的神经响应，因此，对于听神经如何向上层传递声音信息的机理还是当前继续研究的课题。

2.3.3 语音信号听觉模型

听觉系统的研究主要集中在三个方面：听觉系统的实验研究、听觉系统的建模和听觉模型的应用。听觉系统的实验研究主要是指听觉系统在医学、生理学及心理学方面的研究。由于耳蜗深植于颅骨中，尺寸极小（如蜗管的直径只有 1mm），所以耳蜗的实验研究是一项非常艰巨和复杂的工作。

耳蜗建模主要集中在基底膜的振动上，而耳蜗的听觉感受实际上是通过基底膜的振动和毛细胞的转换才能最后变成神经纤维的脉冲发放。然而，建立基底膜的振动模型是耳蜗建模的首要任务，它又被称为耳蜗的宏观力学模型。

目前工程上用得较多的是一种耳蜗的计算模型，它与数学模型不同，它主要是一种算法。其优点是：许多难以在数学模型中得以描述的听觉特性在计算模型中很容易表现出来，它是一种面向应用的耳蜗模型。这里介绍一种计算模型，是 1982 年由美国 Fairchild 人工智能研究室 Lyon 提出的，由三部分组成。第一部分是基底膜的振动模型，它由许多二阶网络组成的串、并联滤波器组构成。由滤波器的总输入到每个滤波器的输出，其传递函数为带通函数，且各相邻滤波器的频率特性高度重叠。这一部分的功能主要是将输入的声音信号在频域上分解，从而在某一部分滤波器的输出端可得到较高信噪比的被分解了的信号输出。第二部分是毛细胞模型，用一个半波整流器加上一个低通滤波器来模拟单个毛细胞的检测功能。半波整流器是用来模拟毛细胞的单向开关特性，由于所采用的是理想半波整流器，所以其后必须用一低通滤波器来消除整流后的高频分量。第三部分是神经纤维模型，这里认为耳蜗神经纤维具有非线性压缩特性，因此，用一种压缩网络模拟神经纤维的这一特点。整个模型共有 64 个通道，系统的输出是一种类似于语谱图的信号。由此得到了听觉模型常用结构图，如图 2.8 所示。

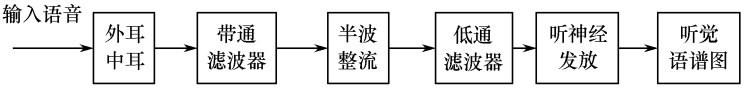


图 2.8 语音信号听觉模型的一般原理框图

后来人们在这些模型的基础上不断改进，也提出了许多其他模型。但即便如此，到目前为止模拟人类的听觉系统仍然很困难，已知的机理知识仍不足以满足工程模型的细节建模要求。由于听觉模型通常包括多级非线性传输级，分析处理变得十分困难，而且大多数模型都依赖于实验数据。因此关于模拟人类的听觉模型进行语音信号的分析依然是一个研究的热点课题。

2.4 语音的感知

2.4.1 几个概念

语音的听觉感知是一个复杂的人脑－心理过程。对听觉感知的研究还很不成熟。听觉感知的试验主要还在测试响度、音高和掩蔽效应等。人耳听觉界限的频率范围大约为 20Hz～

20kHz。在频率范围低端，感觉声音变成低频脉冲串，在高端感觉声音减小直至完全听不到一点声响。语音感知的强度范围是0～130dB声压级（基准声压级为10^{-10} W/cm^2），声音强度太高，感到难以忍受，强度太低则感到寂静无声。

1. 响度

这是频率和强度级的函数。通常用响度（单位为宋）和响度级（单位为方）来表示。

人耳刚刚可以听到的声音强度，称为“听阈”。此时响度级定为零方。测量表明听阈值是随频率变化的。通常，人们把1kHz纯音听阈值定为零方。此时声强为10^{-16} W/cm^2，这样的声波振动几乎不能使鼓膜离开它的静止位置，可见人耳对声音是非常灵敏的。另外，加大声音的强度，使听起来令耳朵感到疼痛，这个阈值称为“痛阈”。测试表明对1kHz的纯音，当声强级大到120dB时，即声强为10^{-4} W/cm^{2}会达到痛阈。可见人耳的听觉范围相当宽，相差10^{12}倍。

响度与响度级是有区别的。60方响度级比30方响度级的声音要响，但没有响了一倍。响度是刻划数量关系的。2宋响度要比1宋响度的声音响一倍。1宋响度被定义为1kHz纯音在声强级为40dB时（声强为10^{-12} W/cm^2）的响度。

2. 音高

音高也称基音。物理单位为赫兹，主观感觉的音高单位是美（Mel）。当声强级为40dB（或响度级为40方）、频率为1kHz时，设定的音高为1000美。

响度与音高之间具有互为补充的关系。例如，可以用频率补充声强使人们感觉到响度相同，也可以用声强补充频率使人感觉音高相同。

2.4.2 掩蔽效应

人耳能感受的频率范围为20Hz～20kHz，其对于频率的分辨能力是非均匀的，在100～500Hz范围内，可分辨的两个纯音的频率之差为$\Delta f \approx 1.8$Hz，而在500Hz～16kHz范围内，相对频率分辨率几乎恒定，$\Delta f/f \approx 3.5\%$，因此，20Hz～20kHz的频率范围总共有620个频率间隔。当然人耳对于频率的分辨能力是受声强影响的，对于太强或太弱的声音，频率分辨率都会降低。人耳对声音的时间分辨力可以短至2ms，这是用两个紧接着的高低不同的声音进行测听，看能否说出是两个音而测得的结果。

两个响度不等的声音作用于人耳时，则响度较高的频率成分的存在会影响到对响度较低的频率成分的感受，使其变得不易察觉，这种现象称为掩蔽效应。由于频率较低的声音在内耳耳蜗基底膜上行波传递的距离大于频率较高的声音，故一般说来，低音容易掩蔽高音，而高音掩蔽低音较难。掩蔽会造成因一个声音的存在，而使另一个声音的听阈上升。

基于上面两点，可以将真实的声音频率映射到“感知”频率尺度上，即Bark尺度对应的临界带宽，于是就引出了临界带宽的概念。

2.4.3 临界带宽与频率群

用一中心频率为f，带宽为Δf的白噪声来掩蔽一频率为f的纯音，先将这个白噪声的强度调节到使被掩蔽纯音恰好听不见为止。然后将Δf由大到小逐渐变化，而保持单位频率的噪声强度（即噪声谱密度）不变，起初这个纯音一直是听不见的，但当Δf小到某个临界值时，这个纯音就突然可以听见了。如果再进一步减小Δf，被掩蔽音f就会越来越清晰。这里刚刚开始能听到被掩蔽声时的Δf宽的频带，叫做频率f处的临界带。当掩蔽噪声的带宽窄于

临界带的带宽时，能掩蔽住纯音 f 的强度是随噪声的带宽的增加而增加的，但当掩蔽噪声的带宽达到临界带后，继续增加噪声带宽就不再引起掩蔽量的提高了。临界带宽是随中心频率而变的，被掩蔽纯音的频率（即临界带的中心频率）越高，临界带宽也越宽。不过二者的变化关系不是一种线性关系。前面已经提到基底膜具有与频谱分析器相似的作用，耳蜗的一个重要功能就是频率分解，不同的频率在沿基底膜的不同位置上集中响应，那么临界频带也可定义为：一个给定的正弦纯音在基底膜上能够产生谐振反应的那一部分。一个频率群的划分相应于基底膜分成许多很小的部分，每一部分对应一个频率群。掩蔽效应就在这些部分内发生，对应同一基底膜的那些频率的声音，在大脑中似乎是叠加在一起进行评价的，如果它们同时发声，可以互相掩蔽，因此，频率群与临界带之间存在密切的联系。一个临界带的单位用巴克（Bark）表示。

2.5 语音信号模型

由 2.2 节介绍的发声机理模型图可知，语音生成系统包含三部分：由声门产生的激励函数 $G(z)$、由声道产生的调制函数 $V(z)$ 和由嘴唇产生的辐射函数 $R(z)$。语音生成系统的传递函数由这三个函数级联而成，即

$$H(z)=G(z)V(z)R(z) \tag{2.1}$$

下面我们将建立这三个函数的数学表达。

2.5.1 激励模型

发浊音时，由于声门不断开启和关闭，产生间隙的脉冲。经仪器测试它类似于斜三角形的脉冲。也就是说，这时的激励波是一个以基音周期为周期的斜三角脉冲串。

如图 2.9 所示为三角波及其频谱图，由程序 2.1 生成。单个三角波的数学表达式为

$$g(n)=\begin{cases}\dfrac{1}{2}\left[1-\cos\dfrac{n\pi}{N_1}\right] & 0\leqslant n\leqslant N_1\\ \cos\left[\dfrac{n-N_1}{2N_2}\pi\right] & N_1\leqslant n\leqslant N_1+N_2\\ 0 & \text{其他}\end{cases} \tag{2.2}$$

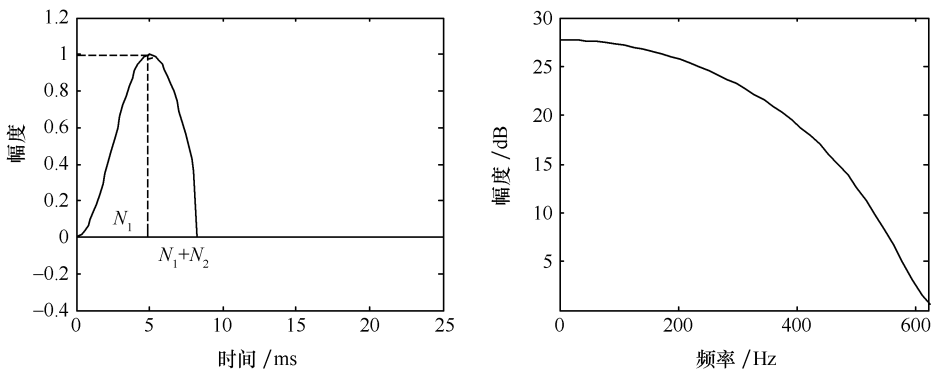


图 2.9 三角波及其频谱图

式中，N_1为斜三角波的上升时间，N_2为其下降时间，由图 2.9 可以看出单个斜三角波的频谱 $G(e^{j\omega})$表现出一个低通滤波器的特性。可以把它表示成 z 变换的全极点形式

$$G(z)=\frac{1}{(1-e^{-cT}\cdot z^{-1})^2} \tag{2.3}$$

这里 c 是一个常数，$T=N_1+N_2$。显然上式表示一个二极点模型。因此，作为激励的斜三角波串可以用一串加了权的单位脉冲序列去激励上述单位斜三角波模型实现。这个单位脉冲串和幅值因子可以表示成下面的 z 变换形式

$$E(z)=\frac{A_v}{1-z^{-1}} \tag{2.4}$$

所以整个激励模型可表示为

$$U(z)=\frac{A_v}{1-z^{-1}}\cdot\frac{1}{(1-e^{-cT}z^{-1})^2} \tag{2.5}$$

在发清音的场合，声道被阻碍形成湍流，所以可以模拟成随机白噪声。

【程序 2.1】sanjiaobopinpu.m

```
% 三角波及其频谱
n=linspace(0,25,125);
g=zeros(1,length(n));
i=0;
for i=0:40
   if n(i+1)<=5
       g(i+1)=0.5*(1-cos(n(i+1)*pi/5));
   else
       g(i+1)=cos((n(i+1)-5)*pi/8);
   end
end
figure(1)
subplot(121)
plot(n,g)
xlabel('时间/ms')
ylabel('幅度')
gtext('N1')
gtext('N1+N2')
axis([0,25,-0.4,1.2])

r=fft(g,1024);                  % 对信号 g 进行 1024 点傅里叶变换
r1=abs(r);                      % 对 r 取绝对值 r1 表示频谱的幅度值
yuanlai=20*log10(r1);           % 对幅值取对数
signal(1:64)=yuanlai(1:64);     % 取 64 个点，目的是画图的时候，维数一致
pinlv=(0:1:63)*8000/512;        % 点和频率的对应关系
subplot(122)
plot(pinlv,signal);
xlabel('频率/Hz')
```

```
ylabel('幅度/dB')
axis([0,620,0,30])
```

2.5.2 声道模型

典型的声道模型有两种，即无损声管模型和共振峰模型。通过两种方法得到的数字模型本质上没有区别。无损声管模型比较复杂，故本节只介绍共振峰模型，关于无损声管模型可参考其他书籍。

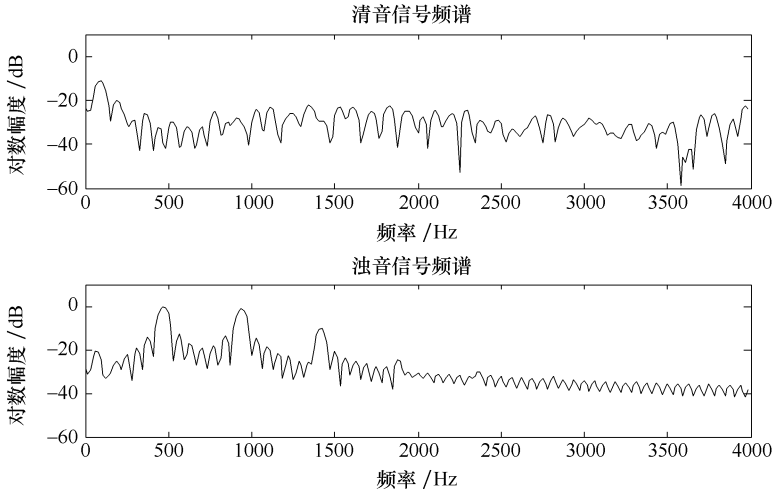


图 2.10　语音信号的频谱

当声波通过声道时，受到声腔共振的影响，在某些频率附近形成谐振。反映在信号频谱图上，在谐振频率处其谱线包络产生峰值，一般把它称做共振峰，如图 2.10 所示。上面的图为清音的频谱图；下面的图为浊音的频谱图，具有明显的峰起，即为共振峰，一般元音可以有 3～5 个共振峰。

从物理声学可以容易推导出均匀断面的共振峰频率。例如，对成人声道 $L=17\text{cm}$ 长，其共振频率计算公式为：$F_i=c(2i-1)/4L$ $i=1,2,3,\cdots,i$ 是共振频率的序号，$c=340\text{m/s}$ 为声速。按此算出前三个共振频率为：$F_1=500\text{Hz}$，$F_2=1500\text{Hz}$，$F_3=2500\text{Hz}$。由于发音时，声道的形状很少是均匀断面的。因此必须通过语音信号来计算共振峰。

一个二阶谐振器的传输函数可以写成

$$V_i(z)=\frac{A_i}{1-B_iz^{-1}-C_iz^{-2}} \tag{2.6}$$

实践表明，用前 3 个共振峰代表一个元音足够了。对于较复杂的辅音或鼻音共振峰的个数要到 5 个以上。多个 V_i 叠加可以得到声道的共振峰模型

$$V(z)=\sum_{i=1}^{M}V_i(z)=\sum_{i=1}^{M}\frac{A_i}{1-B_iz^{-1}-C_iz^{-2}}=\frac{\sum_{r=0}^{R}b_iz^{-r}}{1-\sum_{k=1}^{N}a_kz^{-k}} \tag{2.7}$$

通常 $N>R$，且分子与分母无公共因子及分母无重根。可见，声道模型的传递函数是一个零极点模型，即 ARMA 过程。

语音信号随时间变化的频谱特性可以用语谱图直观地表示。语谱图的纵轴对应于频率，横轴对应于时间，而图像的黑白度对应于信号的能量。所以，声道的谐振频率在图上就表示成为黑带，浊音部分则以出现条纹图形为其特征，这是因为此时的时域波形有周期性，而在浊音的时间间隔内图形显得很致密。图 2.11 为“我到北京去”的语谱图。

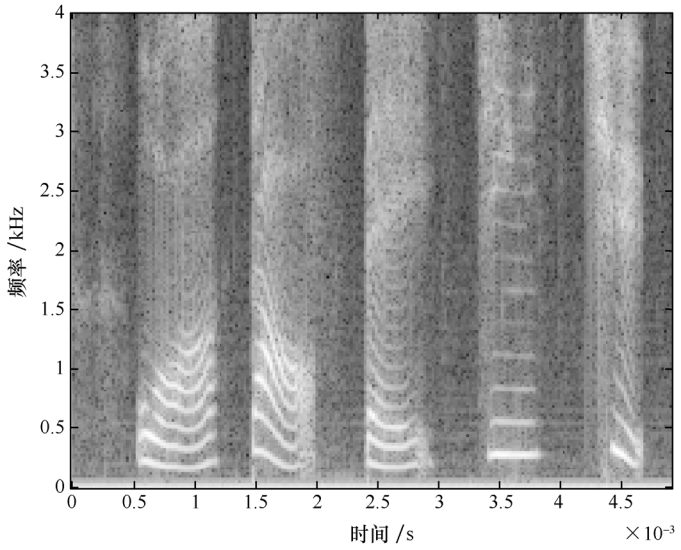


图 2.11 “我到北京去”的语谱图

【程序 2.2】语谱图程序

```
clear all;
[x,sr]=wavread('Beijing.wav');          % sr 为采样频率
if (size(x,1)> size(x,2))               % size(x,1)为 x 的行数,size(x,2)为 x 的列数
    x=x';
end
s=length(x);
w=round(44* sr/1000);                   % 窗长,取离 44* sr/100 最近的整数
n=w;                                    % fft 的点数
ov=w/2;                                 % 50% 的重叠
h=w- ov;
%  win=hanning(n)';                     % 汉宁窗
win=hamming(n)';                        % 汉明窗
c=1;
ncols=1+ fix((s- n)/h);                 % fix 函数是将(s- n)/h 的小数舍去
d=zeros((1+ n/2),ncols);
for b=0:h:(s- n)
    u=win.* x((b+ 1):(b+ n));
    t=fft(u);
    d(:,c)=t(1:(1+ n/2))';
    c=c+ 1;
end
tt=[0:h:(s- n)]/sr;
```

```
ff=[0:(n/2)]* sr/n;
imagesc(tt/1000,ff/1000,20* log10(abs(d)));
colormap(gray);
axis xy
xlabel('时间/s');
ylabel('频率/kHz');
```

2.5.3 辐射模型

从声道模型输出的是速度波，而语音信号是声压波。二者倒比称为辐射阻抗 Z_l，它表征口唇的辐射效应。如果认为口唇张开的面积远远小于头部的表面积，利用单板开槽辐射的处理方法，可以得到辐射阻抗

$$Z_l(\Omega)=\frac{j\Omega L_r R_r}{R_r+j\Omega L_r}=R_0(1-z^{-1}) \tag{2.8}$$

式中

$$R_r=\frac{128}{9\pi^2},\quad L_r=\frac{8a}{3\pi c} \tag{2.9}$$

这里 a 是口唇张开的半径，c 是声波传播速度。由辐射引起的能量损耗正比于辐射阻抗的实部，其频响曲线表现出一阶高通滤波器的特性。在实际信号分析时，常用所谓预加重技术，即在取样之后加入一个一阶高通滤波器。这样，模型只剩下声道部分，对参数分析就方便了。在语音合成时再进行解加重处理。常用的预加重因子为$\left[1-\frac{R(1)}{R(0)}z^{-1}\right]$，这里 $R(n)$是信号 $s(n)$的自相关函数，对浊音 $R(1)/R(0)\approx 1$，对清音该值可取得很小。

2.6 语音信号数字模型

2.6.1 数字模型

前几节分别得到了语音信号激励模型 $G(z)$，辐射模型 $R(z)$和声道模型 $V(z)$，并且知道它们的级联组合形式为 ARMA 模型。这说明语音信号数字模型的传递函数为

$$H(z)=G(z)V(z)R(z)=\frac{\sum_{i=0}^{M}b_i z^{-i}}{\sum_{j=0}^{N}a_j z^{-j}} \tag{2.10}$$

一般情况下，极点个数取 8～12 个，零点个数取 3～5 个，在采样率为 8kHz 或 10kHz 时，$H(z)$在 10～20ms 范围内可以很好地反映语音信号的特征。

根据随机过程理论，一个零点可以用若干极点来近似。因此，适当选取极点个数 p，可以用全极点模型即 AR(p)过程来表达语音信号：

$$H(z)=\frac{G}{1-\sum_{i=1}^{p}a_i z^{-i}} \tag{2.11}$$

在早期 LPC 二元激励模型下，极点个数 p 一般选为 10。对于延时较短或采用后向滤波时，对模型要求较严，必须加入零点或增加极点个数。实际上，对于男声来说，取 20 个极点已

经足够了，考虑女声后，阶数可以加大到 30 阶。语音信号产生的二元激励模型如图 2.12 所示。

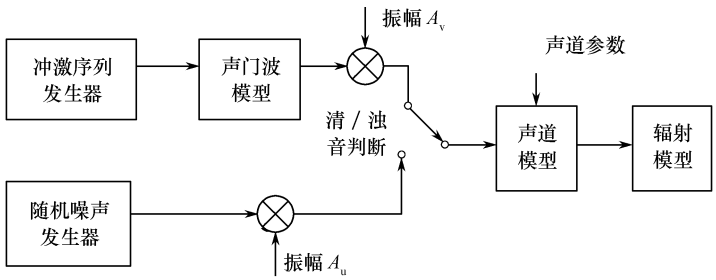


图 2.12　语音信号产生的二元激励模型

2.6.2　模型局限性

声道的传输函数具有全极点的性质，这对于元音和大多数辅音来说是比较符合实际的，但对于鼻音和阻塞音来说，由于出现了零点，这种模型就不够准确了。

一种解决问题的方案是在 $V(z)$ 中引入若干零点，但这将使模型复杂化；另一种方案是适当提高阶数 p，使得全极点模型能更好地逼近具有此种零点的传输函数。数字模型的基本思想是认为任何语音都是由一个适当的激励源作用于声道而产生的，这意味着激励源与声道系统是互相独立的。上述假定对于大多数语音是合适的，但在有些情况下，例如某些瞬变音，实际上声门和声道是互相耦合的，这便形成了这些语音的非线性特性。

并非任何语音都能够明显地按清音和浊音来划分，有些语音甚至也不是清音和浊音的简单叠加。这种将语音信号截然分为周期脉冲激励和噪声激励两种情况的“二元激励”法在高质语音的合成中是不适用的。但二元激励模型，由于其简单性，在早期的语音信号处理研究中使用了许多年。直到 20 世纪 80 年代中期开始，新的激励模型才开始取代二元激励模型。

20 世纪 80 年代中期，人们开始在一个基音周期内采用多个脉冲来构造激励模型。新的激励方法本质上可以归结为存储器模型。就是说将可能的各种激励预先放在存储器内，通过某种判据决定哪一种激励是当前信号的最佳激励，并把这个最佳激励的存储地址作为激励的表征。例如，码激励模型或矢量激励模型等，存储器内容随时间变化的部分称为自适应码书。自适应码书的搜索等价于基音检测。

第3章　语音信号的短时时域分析

3.1　概　　述

语音信号是一种非平稳的时变信号，它携带着各种信息。在语音编码、语音合成、语音识别和语音增强等语音处理中都需要提取语音中包含的各种信息。一般而言语音处理的目的有两种：一种是对语音信号进行分析，提取特征参数，用于后续处理；另一种是加工语音信号，例如在语音增强中对含噪语音进行背景噪声抑制，以获得相对“干净”的语音；在语音合成中需要对分段语音进行拼接平滑，获得主观音质较高的合成语音，这方面的应用同样是建立在分析并提取语音信号信息的基础上的。总之，语音信号分析的目的就在于方便有效地提取并表示语音信号所携带的信息。

根据所分析的参数类型，语音信号分析可以分成时域分析和变换域（频域、倒谱域）分析。其中时域分析方法是最简单、最直观的方法，它直接对语音信号的时域波形进行分析，提取的特征参数主要有语音的短时能量和平均幅度、短时平均过零率、短时自相关函数和短时平均幅度差函数等。本章将介绍这几种时域参数，以及它们在语音信号处理的端点检测和基音周期估值中的应用。

3.2　语音信号的预处理

实际的语音信号是模拟信号，因此在对语音信号进行数字处理之前，首先要将模拟语音信号 $s(t)$ 以采样周期 T 采样，将其离散化为 $s(n)$，采样周期的选取应根据模拟语音信号的带宽（依奈奎斯特采样定理）来确定，以避免信号的频域混叠失真。在对离散后的语音信号进行量化处理过程中会带来一定的量化噪声和失真。实际中获得数字语音的途径一般有两种，正式的和非正式的。正式的是指大公司或语音研究机构发布的被大家认可的语音数据库，非正式的则是研究者个人用录音软件或硬件电路加麦克风随时随地录制的一些发音或语句。通常作为初学者，可使用多媒体计算机，安装相关的音频处理软件即可获得语音数据文件。语音信号的频率范围通常是 300～3400Hz，一般情况下取采样率为 8kHz 即可。本书的数字语音处理对象为语音数据文件，是已经数字化了的语音。

有了语音数据文件后，对语音的预处理包括：预加重和加窗分帧等。

3.2.1　语音信号的预加重处理

对输入的数字语音信号进行预加重，其目的是为了对语音的高频部分进行加重，去除口唇辐射的影响，增加语音的高频分辨率。一般通过传递函数为 $H(z)=1-\alpha z^{-1}$ 的一阶 FIR 高通数字滤波器来实现预加重，其中 α 为预加重系数，$0.9<\alpha<1.0$。设 n 时刻的语音采样值为 $x(n)$，经过预加重处理后的结果为 $y(n)=x(n)-\alpha x(n-1)$，这里取 $\alpha=0.98$。图 3.1 为该高通滤波器的幅频特性和相频特性。图 3.2 中分别给出了预加重前和预加重后的一段浊音信号

及频谱，可以看出，预加重后的频谱在高频部分的幅度得到了提升。实现高频提升的 MATLAB 程序在下面给出。

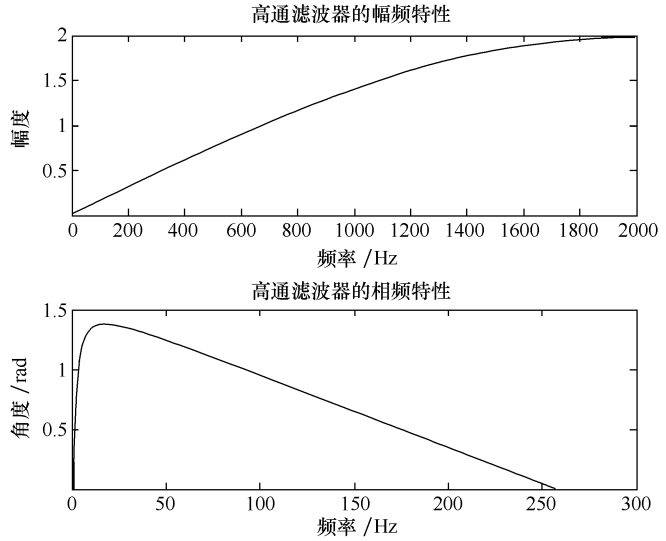


图 3.1　预加重滤波器的幅频特性和相频特性

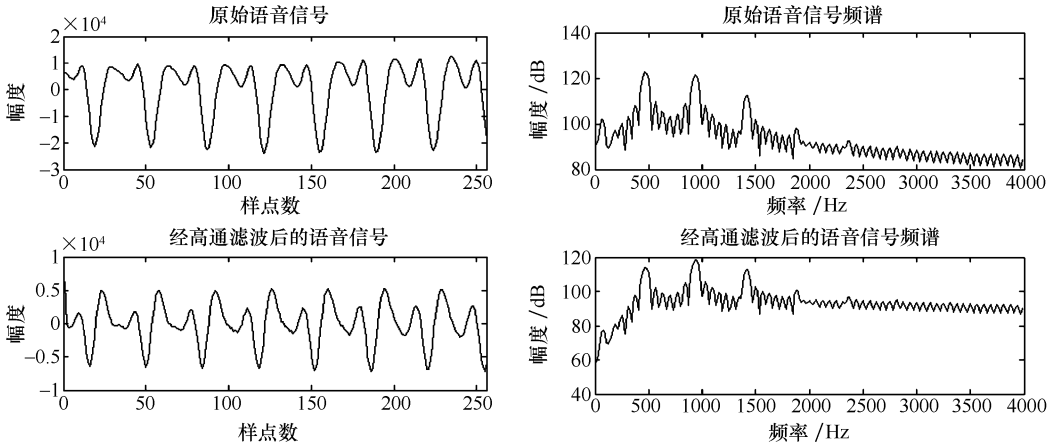


图 3.2　预加重前和预加重后的一段浊音信号及频谱

【程序 3.1】 gaopintisheng.m

```
fid=fopen('voice2.txt','rt')                    % 打开文件
e=fscanf(fid,'% f');                            % 读数据
ee=e(200:455);                                  % 选取原始文件 e 的第 200~ 455 点的语音，
                                                % 也可选其他样点
r=fft(ee,1024);                                 % 对信号 ee 进行 1024 点傅里叶变换
r1=abs(r);                                      % 对 r 取绝对值 r1 表示频谱的幅度值
pinlv=(0:1:255)* 8000/512                       % 点和频率的对应关系
yuanlai=20* log10(r1)                           % 对幅值取对数
signal(1:256)=yuanlai(1:256);                   % 取 256 个点，目的是画图的时候，维数一致
[h1,f1]=freqz([1,- 0.98],[1],256,4000);         % 高通滤波器
```

```
pha=angle(h1);                          % 高通滤波器的相位
H1=abs(h1);                             % 高通滤波器的幅值
r2(1:256)=r(1:256)
u=r2.* h1'                              % 将信号频域与高通滤波器频域相乘相当于
                                          在时域的卷积
u2=abs(u)                               % 取幅度绝对值
u3=20* log10(u2)                        % 对幅值取对数
un=filter([1,- 0.98],[1],ee)            % un 为经过高频提升后的时域信号
figure(1);subplot(211);
plot(f1,H1);title('高通滤波器的幅频特性');
xlabel('频率/Hz');ylabel('幅度');
subplot(212);plot(pha);title('高通滤波器的相频特性');
xlabel('频率/Hz');ylabel('角度/rad');
figure(2);subplot(211);plot(ee);title('原始语音信号');
xlabel('样点数');ylabel('幅度');
axis([0 256 - 3* 10^4 2* 10^4]);
subplot(212);plot(real(un));
title('经高通滤波后的语音信号');axis([0 256 - 1* 10^4 1* 10^4]);
xlabel('样点数');ylabel('幅度');
figure(3);subplot(211);plot(pinlv,signal);title('原始语音信号频谱');
xlabel('频率/Hz');ylabel('幅度/dB');
subplot(212);plot(pinlv,u3);title('经高通滤波后的语音信号频谱');
xlabel('频率/Hz');ylabel('幅度/dB');
```

3.2.2 语音信号的加窗处理

进行预加重数字滤波处理后，接下来进行加窗分帧处理。语音信号是一种随时间而变化的信号，主要分为浊音和清音两大类。浊音的基音周期、清浊音信号幅度和声道参数等都随时间而缓慢变化。由于发声器官的惯性运动，可以认为在一小段时间里（一般为 10～30ms）语音信号近似不变，即语音信号具有短时平稳性。这样，可以把语音信号分为一些短段（称为分析帧）来进行处理。语音信号的分帧是采用可移动的有限长度窗口进行加权的方法来实现的。一般每秒的帧数约为 33～100 帧，视实际情况而定。分帧虽然可以采用连续分段的方法，但一般要采用图 3.3 所示的交叠分段的方法，这是为了使帧与帧之间平滑过渡，保持其连续性。前一帧和后一帧的交叠部分称为帧移，帧移与帧长的比值一般取为 0～1/2，图 3.3 给出了帧移与帧长示意图。

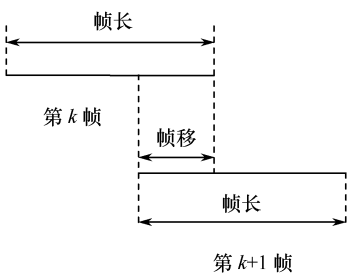


图 3.3　语音信号分帧

常用的窗有两种，一种是矩形窗，窗函数如下：

$$w(n)=\begin{cases}1, & 0\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases} \tag{3.1}$$

另一种是汉明（Hamming）窗，窗函数如下：

$$w(n)=\begin{cases}0.54-0.46\cos[2\pi n/(N-1)], & 0\leqslant n\leqslant N\\ 0, & \text{其他}\end{cases} \tag{3.2}$$

这两种窗的时域和频域波形可用 MATLAB 程序实现，下面分别叙述。

1. 矩形窗时域和频域波形，窗长 $N=61$

【程序 3.2】juxing.m

```
x=linspace(0,100,10001);          % 在 0～100 的横坐标间取 10001 个值
h=zeros(10001,1);                 % 为矩阵 h 赋 0 值
h(1:2001)=0;                      % 前 2000 个值取为 0 值
h(2002:8003)=1;                   % 窗长，窗内值取为 1
h(8004:10001)=0;                  % 后 2000 个值取为 0 值
figure(1);                        % 定义图号
subplot(1,2,1)                    % 画第一个子图
plot(x,h,'k');                    % 画波形，横坐标为 x，纵坐标为 h，k 表示黑色
title('矩形窗时域波形');          % 图标题
xlabel('样点数');                 % 横坐标名称
ylabel('幅度');                   % 纵坐标名称
axis([0,100,- 0.5,1.5])           % 限定横、纵坐标范围
line([0,100],[0,0])               % 画出 x 轴
w1=linspace(0,61,61);             % 取窗长内的 61 个点
w1(1:61)=1;                       % 赋值 1，相当于矩形窗
w2=fft(w1,1024);                  % 对时域信号进行 1024 点的傅里叶变换
w3=w2/w2(1)                       % 幅度归一化
w4=20* log10(abs(w3));            % 对归一化幅度取对数
w=2* [0:1023]/1024;               % 频率归一化
subplot(1,2,2);                   % 画第二个子图
plot(w,w4,'k')                    % 画幅度特性图
axis([0,1,- 100,0])               % 限定横、纵坐标范围
title('矩形窗幅度特性');          % 图标题
xlabel('归一化频率 f/fs');        % 横坐标名称
ylabel('幅度/dB');                % 纵坐标名称
```

图 3.4 为程序运行后相应的矩形窗时域波形和幅频特性图。

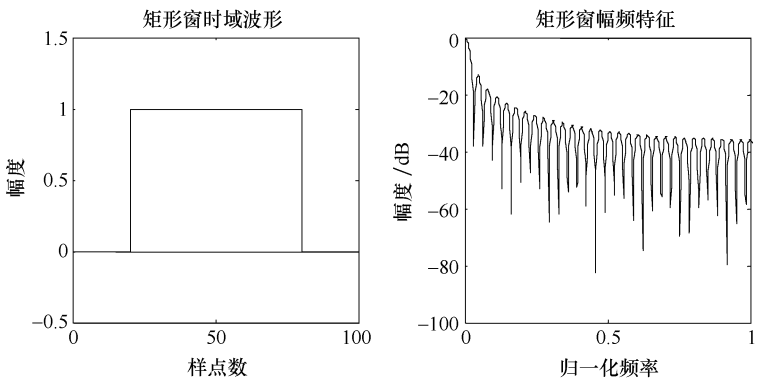


图 3.4　矩形窗及其频谱

2. 汉明(Hamming)窗时域和频域波形,窗长 $N=61$

【程序 3.3】haming.m

```
x=linspace(20,80,61);                % 在 20～80 的横坐标间取 61 个值作为横坐标点
h=hamming(61);                       % 取 61 个点的汉明窗值为纵坐标值
figure(1);                           % 画图
subplot(1,2,1);                      % 第一个子图
plot(x,h,'k');                       % 横坐标为 x,纵坐标为 h,k 表示黑色
title('汉明窗时域波形');              % 图标题
xlabel('样点数'); ylabel('幅度');     % 横纵坐标名称
w1=linspace(0,61,61);                % 取窗长内的 61 个点
w1(1:61)=hamming(61);                % 加汉明窗
w2=fft(w1,1024);                     % 对时域信号进行 1024 点傅里叶变换
w3=w2/w2(1);                         % 幅度归一化
w4=20* log10(abs(w3))                % 对归一化幅度取对数
w=2* [0:1023]/1024;                  % 频率归一化
subplot(1,2,2)                       % 画第二个子图
plot(w,w4,'k')                       % 画幅度特性图
axis([0,1,- 100,0])                  % 限定横、纵坐标范围
title('汉明窗幅度特性');              % 图标题
xlabel('归一化频率); ylabel('幅度/dB');   % 横纵坐标名称
```

图 3.5 为程序运行后相应的汉明窗时域波形和幅频特性。

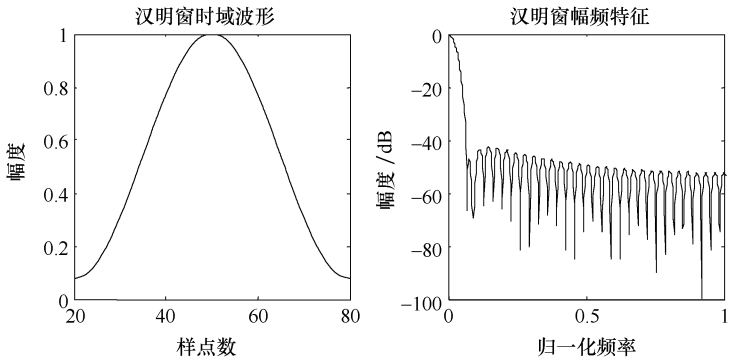


图 3.5 汉明窗及其频谱

对比图 3.4 与图 3.5 可以看出,矩形窗的主瓣宽度小于汉明窗,具有较高的频谱分辨率,但是矩形窗的旁瓣峰值较大,因此其频谱泄漏比较严重。相比较,虽然汉明窗的主瓣宽度较宽,约大于矩形窗的一倍,但是它的旁瓣衰减较大,具有更平滑的低通特性,能够在较高的程度上反映短时信号的频率特性。

图 3.6 说明了加窗方法,其中窗序列沿着语音样点值序列 $x(n)$ 逐帧从左向右移动,窗 $w(n)$ 长度为 N。

在确定了窗函数以后,对语音信号的分帧处理,实际上就是对各帧进行某种变换或运算。设这种变换或运算用 $T[\ ]$ 表示,$x(n)$ 为输入语音信号,$w(n)$ 为窗序列,$h(n)$ 是与 $w(n)$ 有关的滤波器,则各帧经处理后的输出可以表示为

$$Q_n = \sum_{m=-\infty}^{\infty} T[x(m)]h(n-m) \tag{3.3}$$

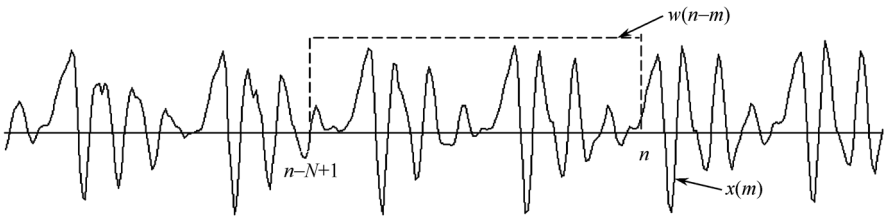


图 3.6　加窗方法示意图

几种常见的短时处理方法是：

1. $T[x(m)]=x^2(m)$，$h(n)=w^2(n)$，Q_n 对应于能量。

2. $T[x(m)]=|\mathrm{sgn}[x(m)]-\mathrm{sgn}[x(m-1)]|$，$h(n)=w(n)$，$Q_n$ 对应于平均过零率。

3. $T[x(m)]=x(m)x(m+k)$，$h(n)=w(n)w(n+k)$，Q_n 对应于自相关函数。

3.3　短时平均能量

由于语音信号的能量随时间而变化，清音和浊音之间的能量差别相当显著。因此对短时能量和短时平均幅度进行分析，可以描述语音的这种特征变化情况。

定义 n 时刻某语音信号的短时平均能量 E_n 为

$$E_n = \sum_{m=-\infty}^{+\infty} [x(m)w(n-m)]^2 = \sum_{m=n-(N-1)}^{n} [x(m)w(n-m)]^2 \tag{3.4}$$

式中，N 为窗长，可见短时能量为一帧样点值的加权平方和。特殊地，当窗函数为矩形窗时，有

$$E_n = \sum_{m=n-(N-1)}^{n} x^2(m) \tag{3.5}$$

也可以从另外一个角度来解释。令

$$h(n)=w^2(n) \tag{3.6}$$

式(3.4)可以表示为

$$E_n = \sum_{m=-\infty}^{+\infty} x^2(m)h(n-m) = x^2(n) * h(n) \tag{3.7}$$

式(3.7)可以理解为：首先语音信号各个样点值平方，然后通过一个冲激响应为 $h(n)$ 的滤波器，输出为由短时能量构成的时间序列，如图 3.7 所示。

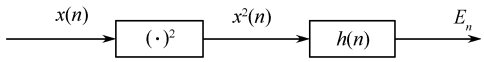


图 3.7　语音信号的短时平均能量实现框图

冲激响应 $h(n)$ 的选择或者说窗函数的选择直接影响着短时能量的计算。若 $h(n)$ 幅度恒定，其序列长度 N(即窗长)很长，这样的窗等效为很窄的低通滤波器，此时 $h(n)$ 对 $x^2(n)$ 的平滑作用非常显著，使得短时能量几乎没多大变化，无法反映语音的时变特性。反之，若 $h(n)$ 序列长度 N 过小，那么等效窗又不能提供足够的平滑，以至于语音振幅瞬时变化的许多细节仍然被保留了下来，从而看不出振幅包络的变化规律。通常 N 的选择与语音的基音周期相联

系，一般要求窗长为几个基音周期的数量级。由于语音基音频率范围为50～500Hz，因此折中选择帧长为10～20ms。图3.8画出了一段实际语音（女声“我到北京去”）的短时能量函数随矩形窗长的变化曲线，横坐标为帧数，帧间无交叠。图中的4幅图分别对应 $N=50$，$N=100$，$N=400$，$N=800$。从图中可以看到，$N=50$ 和 $N=100$ 的短时平均能量曲线不够平滑，而 $N=800$ 的曲线又过于平滑，将个别的细节变化平滑掉了；$N=400$ 的曲线比较合适。下面给出产生该图的MATLAB程序实现过程。

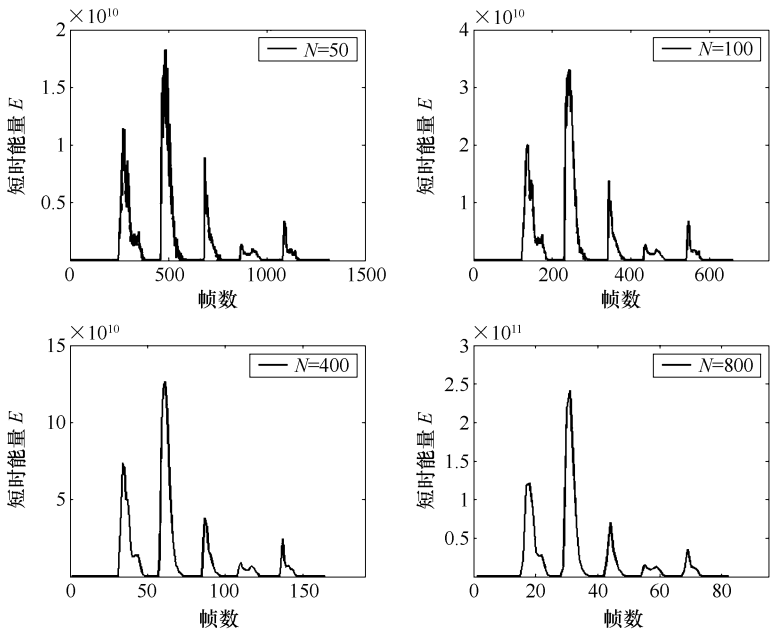


图3.8　不同矩形窗长 N 时的短时能量函数

将读入的语音wav文件保存为txt文件，设置采样率为8kHz，16位，单声道。

【程序3.4】nengliang.m

```
fid=fopen('zqq.txt','rt');                % 读入语音文件
x=fscanf(fid,'% f');
fclose(fid);
% 计算 N=50，帧移=50 时的语音能量
s=fra(50,50,x)                            % 对输入的语音信号进行分帧，其中帧长 50，帧移 50
s2=s.^2;                                  % 一帧内各样点的能量
energy=sum(s2,2)                          % 求一帧能量
subplot(2,2,1)                            % 定义画图数量和布局
plot(energy)                              % 画 N=50 时的语音能量图
xlabel('帧数')                            % 横坐标
ylabel('短时能量 E')                      % 纵坐标
legend('N=50')                            % 曲线标识
axis([0,1500,0,2* 10^10])                 % 定义横纵坐标范围
% 计算 N=100，帧移=100 时的语音能量
s=fra(100,100,x)
s2=s.^2;
```

```
energy=sum(s2,2)
subplot(2,2,2)
plot(energy)                          % 画 N=100 时的语音能量图
xlabel('帧数')
ylabel('短时能量 E')
legend('N=100')
axis([0,750,0,4* 10^10])              % 定义横纵坐标范围
% 计算 N=400,帧移=400 时的语音能量
s=fra(400,400,x)
s2=s.^2;
energy=sum(s2,2)
subplot(2,2,3)
plot(energy)                          % 画 N=400 时的语音能量图
xlabel('帧数')
ylabel('短时能量 E')
legend('N=400')
axis([0,190,0,1.5* 10^11])            % 定义横纵坐标范围
% 计算 N=800,帧移=800 时的语音能量
s=fra(800,800,x)
s2=s.^2;
energy=sum(s2,2)
subplot(2,2,4)
plot(energy)                          % 画 N=800 时的语音能量图
xlabel('帧数')
ylabel('短时能量 E')
legend('N=800')
axis([0,95,0,3* 10^11])               % 定义横纵坐标范围
```

其中 fra()为分帧函数,其 MATLAB 程序如下:

```
%  fra.m
function f=fra(len,inc,x)             % 对读入语音分帧,len 为帧长;inc 为帧重叠样点
                                      % 数;x 为输入语音数据
fh=fix(((size(x,1)- len)/inc)+ 1)     % 计算帧数
f=zeros(fh,len);                      % 设一个零矩阵,行为帧数,列为帧长
i=1;n=1;
while i< =fh                          % 帧间循环
    j=1;
    while j< =len                     % 帧内循环
        f(i,j)=x(n);
        j=j+ 1;n=n+ 1;
    end
    n=n- len+ inc;                    % 下一帧开始位置
    i=i+ 1;
    end
```

短时平均能量的主要用途如下:

① 可以作为区分清音和浊音的特征参数。实验结果表明浊音的能量明显高于清音。通过设置一个能量门限值，可以大致判定浊音变为清音或者清音变为浊音的时刻，同时可以大致划分浊音区间和清音区间。

② 在信噪比较高的情况下，短时能量还可以作为区分有声和无声的依据。

③ 可以作为辅助的特征参数用于语音识别中。

3.4 短时平均幅度函数

短时能量的一个主要问题是 E_n 对信号电平值过于敏感。由于需要计算信号样值的平方和，在定点实现时很容易产生溢出。为了克服这个缺点，可以定义一个短时平均幅度函数 M_n 来衡量语音幅度的变化：

$$M_n = \sum_{m=-\infty}^{+\infty} |x(m)|w(n-m) = \sum_{m=n-N+1}^{n} |x(n)|w(n-m) \tag{3.8}$$

式(3.8)可以理解为 $w(n)$ 对 $|x(n)|$ 的线性滤波运算，实现框图如图 3.9 所示。与短时能量比较，短时平均幅度相当于用绝对值之和代替了平方和，简化了运算。

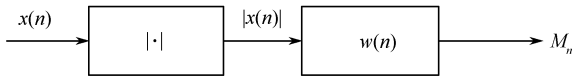


图 3.9 短时平均幅度实现框图

图 3.10 画出了短时平均幅度函数随矩形窗窗长 N 变化的情况，帧间无交叠。比较图 3.8 和图 3.10，窗长 N 对平均幅度函数的影响与短时能量的分析结论是完全一致的。但由于平均幅度函数没有平方运算，因此其动态范围(最大值与最小值之差)要比短时能量小，接近于标准能量计算的动态范围的平方根。所以，尽管短时平均幅度也可以用来区分清音和浊音、无声和有声，但是二者之间的幅度差就不如短时能量那么明显。

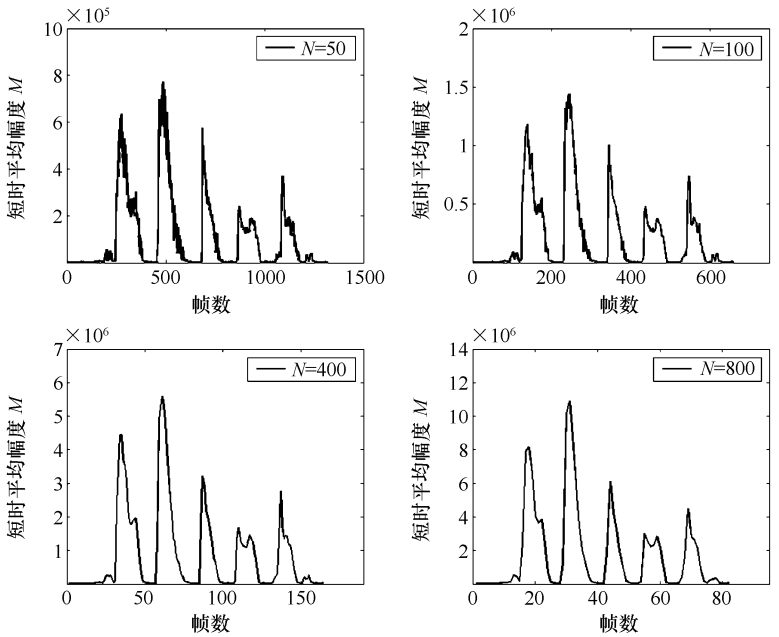


图 3.10 不同矩形窗长 N 时的短时平均幅度函数

其 MATLAB 的具体实现如下，其中每行程序的意义可参见短时平均能量的解释。

【程序 3.5】fudu.m

```
fid=fopen('zqq.txt','rt')                    % 读入语音文件
x=fscanf(fid,'% f')
fclose(fid)

s=fra(50,50,x)                               % 语音短时平均幅度图
s3=abs(s)
avap=sum(s3,2)
subplot(2,2,1)
plot(avap)
xlabel('帧数')
ylabel('短时平均幅度 M')
legend('N=50')
axis([0,1500,0,10* 10^5])
s=fra(100,100,x)
s3=abs(s)
avap=sum(s3,2)
subplot(2,2,2)
plot(avap)
xlabel('帧数')
ylabel('短时平均幅度 M')
legend('N=100')
axis([0,750,0,2* 10^6])
s=fra(400,400,x)
s3=abs(s)
avap=sum(s3,2)
subplot(2,2,3)
plot(avap)
xlabel('帧数')
ylabel('短时平均幅度 M')
legend('N=400')
axis([0,190,0,7* 10^6])

s=fra(800,800,x)
s3=abs(s)
avap=sum(s3,2)
subplot(2,2,4)
plot(avap)
xlabel('帧数')
ylabel('短时平均幅度 M')
legend('N=800')
axis([0,95,0,14* 10^6])
```

3.5 短时平均过零率

短时平均过零率是语音信号时域分析中的一种特征参数。它是指每帧内信号通过零值的次数。对有时间横轴的连续语音信号，可以观察到语音的时域波形通过横轴的情况。在离散时间语音信号情况下，如果相邻的采样具有不同的代数符号就称为发生了过零，因此可以计算过零的次数。单位时间内过零的次数就称为过零率。一段长时间内的过零率称为平均过零率。如果是正弦信号，其平均过零率就是信号频率的两倍除以采样频率，而采样频率是固定的。因此过零率在一定程度上可以反映信号的频率信息。语音信号不是简单的正弦序列，所以平均过零率的表示方法就不那么确切。但由于语音是一种短时平稳信号，采用短时平均过零率仍然可以在一定程度上反映其频谱性质，由此可获得谱特性的一种粗略估计。短时平均过零率的定义为

$$\begin{aligned} Z_n &= \sum_{m=-\infty}^{+\infty} |\operatorname{sgn}[x(m)]-\operatorname{sgn}[x(m-1)]|\, w(n-m) \\ &= |\operatorname{sgn}[x(n)]-\operatorname{sgn}[x(n-1)]| * w(n) \end{aligned} \tag{3.9}$$

其中，sgn[ ]为符号函数，即

$$\operatorname{sgn}[x(n)]=\begin{cases} 1, & x(n)\geqslant 0 \\ -1, & x(n)<0 \end{cases} \tag{3.10}$$

$w(n)$为窗函数，计算时常采用矩形窗，窗长为 N。可以这样理解：当相邻两个样点符号相同时，$|\operatorname{sgn}[x(m)]-\operatorname{sgn}[x(m-1)]|=0$，没有产生过零；当相邻两个样点符号相反时，$|\operatorname{sgn}[x(m)]-\operatorname{sgn}[x(m-1)]|=2$，为过零次数的 2 倍。因此在统计一帧（$N$ 点）的短时平均过零率时，求和后必须要除以 $2N$。这样就可以将窗函数 $w(n)$表示为

$$w(n)=\begin{cases} \dfrac{1}{2N}, & 0\leqslant n\leqslant N-1 \\ 0, & \text{其他} \end{cases} \tag{3.11}$$

在矩形窗条件下，式(3.11)可以简化为

$$Z_n=\frac{1}{2N}\sum_{m=n-(N-1)}^{n}|\operatorname{sgn}[x(m)]-\operatorname{sgn}[x(m-1)]| \tag{3.12}$$

按照式(3.9)，可得出实现短时平均过零率的运算图，如图 3.11 所示。

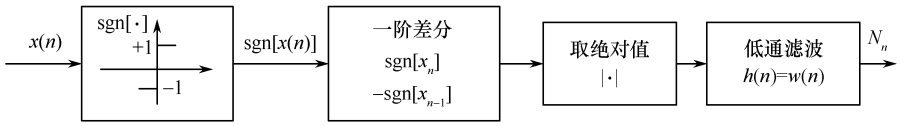


图 3.11 语音信号的短时平均过零率

图 3.12 画出了语音（女声“我到北京去”）的短时平均过零次数的变化曲线，图中窗长 $N=220$，帧重叠 50%。从图中可以看出清音与浊音的短时过零率区别还是比较明显的。

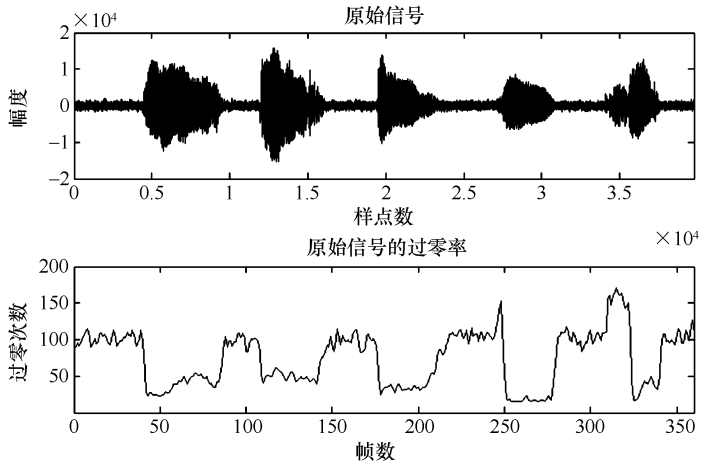


图 3.12　一句语音的短时平均过零率

【程序 3.6】guoling.m

```
clear all
fid=fopen('beijing.txt','rt')
x1=fscanf(fid,'% f');
fclose(fid);
x=awgn(x1,15,'measured');% 加入 15dB 的噪声
s=fra(220,110,x);% 分帧,帧移 110
zcr=zcro(s);% 求过零率
figure(1);
subplot(2,1,1)
plot(x);
title('原始信号');
xlabel('样点数');
ylabel('幅度');
axis([0,39760,- 2* 10^4,2* 10^4]);
subplot(2,1,2)
plot(zcr);
title('原始信号的过零率');
xlabel('帧数');
ylabel('过零次数');
axis([0,360,0,200]);
```

其中 zcro( )为求过零率的函数,其 MATLAB 程序如下:

```
% zcro.m
function f=zcro(x)
f=zeros(size(x,1),1);      % 生成全零矩阵
for i=1:size(x,1)
   z=x(i,:);                          % 提取一行数据
   for j=1:(length(z)- 1);
      if z(j)* z(j+ 1)< 0;
```

```
            f(i)=f(i)+ 1;
        end
    end
end
```

短时平均过零率可以用于语音信号清、浊音的判断。语音产生模型表明，由于声门波引起了谱的高频跌落，所以浊音语音能量约集中在 3kHz 以下。但对于清音语音，多数能量却是出现在较高的频率上。所以，如果过零率高，语音信号就是清音，如果过零率低，语音信号就是浊音。但有的音，位于浊音和清音的重叠区域，这时，只根据短时平均过零率就不可能来明确地判别清、浊音。

端点检测是语音信号处理中的一个基本问题，其目的是从包含语音的一段信号中确定出语音的起点及结束点。有效的端点检测不仅能使处理时间减到最少，而且能抑制无声段的噪声干扰，提高语音处理的质量。有些发音仅用过零率来判断其起点和终点是比较困难的，包括下面几种情况：

- 开始和末尾是弱摩擦音(f，th，h)
- 开始和末尾是弱爆破音(p，t，k)
- 末尾是鼻音
- 浊擦音在字的终了变为清音
- 在一个发音的终止为拖长的元音

当遇到上述情况时，端点检测发生困难，这时可把短时能量和过零率结合起来使用，也可以使用其他改进方法。

3.6 短时自相关分析

3.6.1 短时自相关函数

自相关函数用于衡量信号自身时间波形的相似性。由前面的讨论可知，清音和浊音的发声机理不同，因而在波形上也存在着较大的差异。浊音的时间波形呈现出一定的周期性，波形之间相似性较好；清音的时间波形呈现出随机噪声的特性，杂乱无章，样点间的相似性较差。这样，可以用短时自相关函数来测定语音的相似特性。

时域离散确定信号的自相关函数定义为

$$R(k) = \sum_{m=-\infty}^{+\infty} x(m)x(m+k) \tag{3.13}$$

时域离散随机信号的自相关函数定义为

$$R(k) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{m=-N}^{N} x(m)x(m+k) \tag{3.14}$$

若信号为一周期信号，周期为 P，则

$$R(k)=R(k+P) \tag{3.15}$$

上式说明，周期信号的自相关函数也是一个同样周期的周期信号，自相关函数具有下述性质：

① 对称性 $R(k)= R(-k)$。

② 在 $k=0$ 处为最大值，即对于所有 k 来说，$|R(k)|\leqslant R(0)$。

③ 对于确定信号，值 $R(0)$ 对应于能量，而对于随机信号，$R(0)$ 对应于平均功率。

在上述的第②个性质中，如果是一个周期为 P 的信号，则在取样 $0,\pm P,\pm 2P,\cdots$ 处，其自相关函数也是最大值，因此可以根据自相关函数的最大值的位置来估计周期信号的周期值。

3.6.2 语音信号的短时自相关函数

对于语音来说，采用短时分析方法，可以定义短时自相关函数为

$$R_n(k)=\sum_{m=-\infty}^{+\infty}x(m)w(n-m)x(m+k)w(n-k-m) \tag{3.16}$$

因为 $R_n(-k)=R_n(k)$，所以

$$R_n(k)=R_n(-k)=\sum_{m=-\infty}^{+\infty}[x(m)x(m-k)][w(n-m)w(n-m+k)] \tag{3.17}$$

定义

$$h_k(n)=w(n)w(n+k) \tag{3.18}$$

那么式(3.16)可以写成

$$R_n(k)=\sum_{m=-\infty}^{+\infty}x(m)x(m-k)h_k(n-m) \tag{3.19}$$

式(3.19)表明，序列 $x(n)x(n-k)$ 经过一个冲激响应为 $h_k(n)$ 的数字滤波器滤波即得到短时自相关函数 $R_n(k)$，如图 3.13 所示。

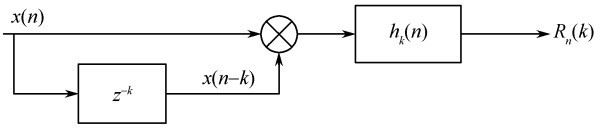


图 3.13 短时自相关函数的框图表示

也可采用直接运算的方法，令 $m=n+m'$，代入式(3.16)中，且令 $w(-m)=w'(m)$，则可得

$$\begin{aligned}R_n(k)&=\sum_{m'=-\infty}^{+\infty}[x(n+m')w(-m')][x(n+m'+k)w'(k+m')]\\&=\sum_{m=-\infty}^{+\infty}[x(n+m)w'(m)][x(n+m+k)w'(k+m)]\end{aligned} \tag{3.20}$$

注意：当 $0\leqslant m\leqslant N-1$ 时，$w'(m)$ 为非零值；当 $0\leqslant k+m\leqslant N-1$ 或 $-k\leqslant m\leqslant N-1-k$ 时，$w'(k+m)$ 为非零值，故 $w'(m)$ 和 $w'(k+m)$ 均为非零值时，则为 $0\leqslant m\leqslant N-1-k$，故式(3.20)可以写成

$$R_n(k)=\sum_{m=0}^{N-1-k}[x(n+m)w'(m)][x(n+m+k)w'(k+m)] \tag{3.21}$$

上式这种直接计算 $R_n(k)$ 的运算量较大，可用 FFT 法来减小运算量。

图 3.14 和图 3.15 分别给出了浊音和清音的短时自相关函数曲线，分别画出了时域波形、加矩形窗和加汉明窗后用式(3.21)计算短时自相关归一化后的结果。语音的抽样频率为 8kHz，窗长为 320。

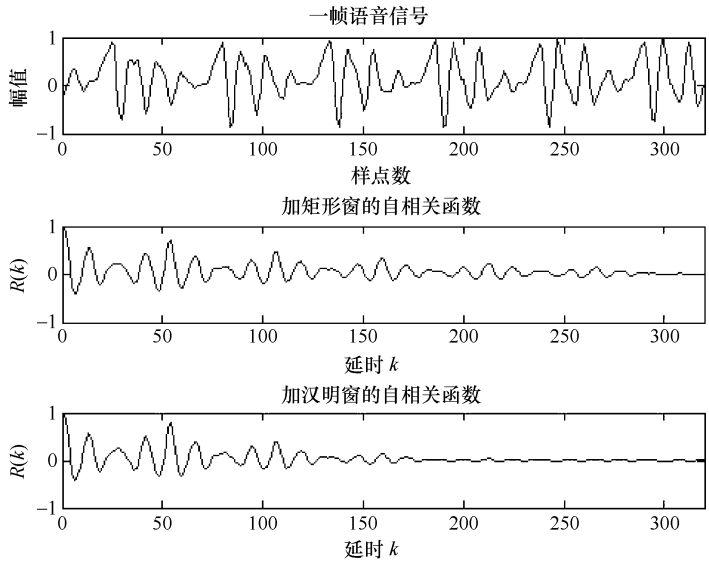


图 3.14　浊音的短时自相关函数

【程序 3.7】zhuoyinzixiangguan.m

```
fid=fopen('voice.txt','rt')
x=fscanf(fid,'% f');
fclose(fid);

s1=x(1:320);                          % 选择一段 320 点的语音段
N=320;                                % 选择的窗长

A=[];                                 % 加 N=320 的矩形窗
for k=1:320;
sum=0;
for m=1:N- k+ 1;
sum=sum+ s1(m)* s1(m+ k- 1);          % 计算自相关
end
A(k)=sum;
end
for k=1:320
A1(k)=A(k)/A(1);                      % 归一化 A(k);
end

f=zeros(1,320);                       % 加 N=320 的汉明窗
n=1;j=1;
   while j< =320
      f(1,j)=x(n)* [0.54- 0.46* cos(2* pi* n/319)];
      j=j+ 1;n=n+ 1;
   end
B=[];
```

```
for k=1:320;
sum=0;
for m=1:N- k+ 1;
sum=sum+ f(m)* f(m+ k- 1);
end
B(k)=sum;
end
for k=1:320
B1(k)=B(k)/B(1);                                    % 归一化 B(k)
end
s2=s1/max(s1);
figure(1)
subplot(3,1,1)
plot(s2)
title('一帧语音信号')
xlabel('样点数')
ylabel('幅值')
axis([0,320,- 1,1]);
subplot(3,1,2)
plot(A1);
title('加矩形窗的自相关函数')
xlabel('延时 k')
ylabel('R(k)')
axis([0,320,- 1,1]);
subplot(3,1,3)
plot(B1);
title('加汉明窗的自相关函数')
xlabel('延时 k')
ylabel('R(k)')
axis([0,320,- 1,1]);
```

清音的短时自相关函数 MATLAB 程序的实现与浊音的基本一致，需要改动的地方只是文件名及显示图形时浊音波形的动态范围。故这里不再给出详细程序。

从图 3.14 和图 3.15 中，可以看出浊音和清音的短时自相关函数有如下几个特点：

① 短时自相关函数可以很明显的反映出浊音信号的周期性。

② 清音的短时自相关函数没有周期性，也不具有明显突出的峰值，其性质类似于噪声。

③ 不同的窗对短时自相关函数结果有一定的影响。采用矩形窗时，浊音自相关曲线的周期性显示出比用汉明窗时更明显的周期性。其主要原因是加汉明窗后，语音段两端的幅度逐渐下降，从而模糊了信号的周期性。

窗长对浊音的短时自相关性有着直接的影响。一方面，由于语音信号的特性是变化的，因此要求 N 应尽量小。但与之相矛盾的另一方面是为了充分反映语音的周期性，又必须选择足够宽的窗，以使得选出的语音段包含两个以上的基音周期。由于基音频率的分布在50～500Hz的范围内，8kHz 采样时对应于 16～160 点，那么窗长 N 的选择要求 $N \geqslant 320$。如图 3.16所示，分别用 $N=320$，$N=160$，$N=70$ 的矩形窗对图 3.14 的浊音段加窗。当$N=70$

时由于窗长不足两个基音周期，所以将不能正确检测基音周期。从图 3.16 也可看到，采用式(3.21)计算出来的短时自相关函数，其幅度是一个逐渐衰减的曲线。这是由于在计算短时自相关时，窗选语音段为有限长度 N，而求和上限为 $N-1-k$，因此当 k 增加时可用于计算的数据就越来越少了，从而导致 k 增加时自相关函数的幅度减小。

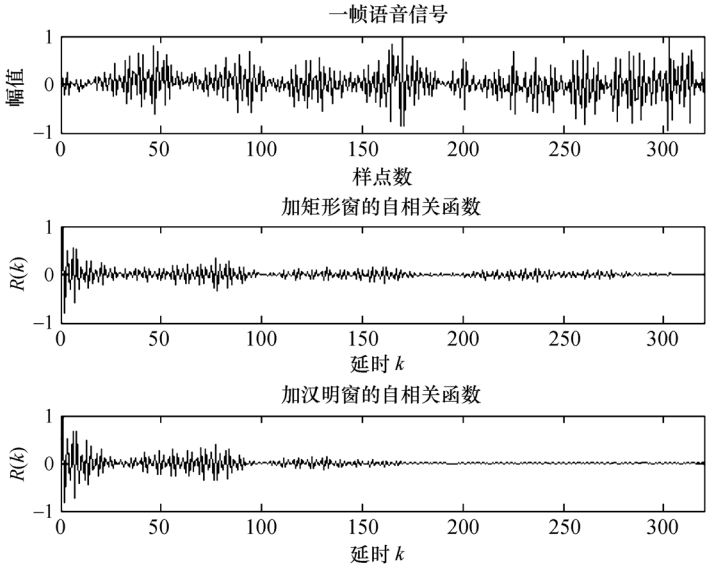


图 3.15　清音的短时自相关函数

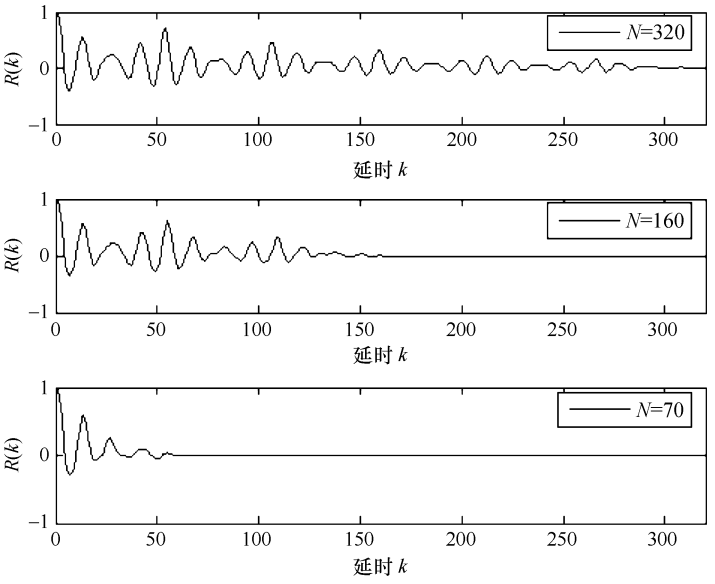


图 3.16　不同矩形窗长时的短时自相关函数

【程序 3.8】duanshizixiangguan.m

```
fid=fopen('voice.txt','rt')
x=fscanf(fid,'% f');
fclose(fid);
s1=x(1:320);
```

```
N＝320;                                  % 选择的窗长,加 N＝320 的矩形窗
A＝[];
for k＝1:320;
sum＝0;
for m＝1:N- (k- 1);
sum＝sum+ s1(m)* s1(m+ k- 1);            % 计算自相关
end
A(k)＝sum;
end
for k＝1:320
A1(k)＝A(k)/A(1);                        % 归一化 A(k);
end
N＝160;                                  % 选择的窗长，% 加 N＝160 的矩形窗
B＝[];
for k＝1:320;
sum＝0;
for m＝1:N- (k- 1);
sum＝sum+ s1(m)* s1(m+ k- 1);            % 计算自相关
end
B(k)＝sum;
end
for k＝1:320
B1(k)＝B(k)/B(1);                        % 归一化 B(k);
end
N＝70;                                   % 选择的窗长,加 N＝70 的矩形窗
C＝[];
for k＝1:320;
sum＝0;
for m＝1:N- (k- 1);
sum＝sum+ s1(m)* s1(m+ k- 1);            % 计算自相关
end
C(k)＝sum;
end
for k＝1:320
C1(k)＝C(k)/C(1);                        % 归一化 C(k);
end
figure(1)
subplot(3,1,1)
plot(A1)
xlabel('延时 k')
ylabel('R(k)')
axis([0,320,- 1,1]);
legend('N＝320')
subplot(3,1,2)
```

```
plot(B1);
xlabel('延时 k')
ylabel('R(k)')
axis([0,320,- 1,1]);
legend('N＝160')
subplot(3,1,3)
plot(C1);
xlabel('延时 k')
ylabel('R(k)')
axis([0,320,- 1,1]);
legend('N＝70')
```

根据上面的分析，如果长基音周期用窄的窗，将得不到预期的基音周期；但是如果是短的基音周期用长的窗，自相关函数将对多个基音周期作平均计算，从而模糊语音的短时特性，这是不希望的。最理想的方法是让窗长自适应于基音周期的变化，但这样会增加计算复杂度。为了解决这个问题，可以采用修正的短时自相关函数，这种方法可以采用较窄的窗，同时避免了短时自相关函数随 k 增加而衰减的不足。

3.6.3 修正的短时自相关函数

修正的短时自相关函数，其定义如下：

$$\hat{R}_n(k)=\sum_{m=-\infty}^{+\infty}x(m)w_1(n-m)x(m+k)w_2(n-m-k) \tag{3.22}$$

若令 $m=n+m'$，代入式(3.22)中，可得

$$\hat{R}_n(k)=\sum_{m'=-\infty}^{+\infty}x(n+m')w_1(-m')x(n+m'+k)w_2(-m'-k) \tag{3.23}$$

定义

$$\begin{cases}\hat{w}_1(m)=w_1(-m)\\ \hat{w}_2(m)=w_2(-m)\end{cases}$$

则有

$$\hat{R}_n(k)=\sum_{m=-\infty}^{+\infty}x(n+m)\hat{w}_1(m)x(n+m+k)\hat{w}_2(m+k) \tag{3.24}$$

$$\begin{aligned}\hat{w}_1(m)&=\begin{cases}1, & 0\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases}\\ \hat{w}_2(m)&=\begin{cases}1, & 0\leqslant n\leqslant N-1+K\\ 0, & \text{其他}\end{cases}\end{aligned} \tag{3.25}$$

式中，K 为 k 的最大值，即 $0\leqslant k\leqslant K$。

由式(3.25)可知，要使 $\hat{w}_2(m+k)$ 为非零值，必须使 $m+k\leqslant N-1+K$，考虑到 $k\leqslant K$，可得 $m\leqslant N-1$，故式(3.24)可以写成

$$\hat{R}_n(k) = \sum_{m=0}^{N-1} x(n+m)x(n+m+k) \tag{3.26}$$

【程序 3.9】xiuzhengzixiangguan.m

```
fid=fopen('voice.txt','rt')
b=fscanf(fid,'% f');

b1=b(1:640);
N=320;                                  % 选择的窗长
A=[];
for k=1:320;
sum=0;
for m=1:N;
sum=sum+ b1(m)* b1(m+ k- 1);
end
A(k)=sum;
end
for k=1:320
A1(k)=A(k)/A(1);                        % 归一化
end
figure(1)
subplot(3,1,1)
plot(A1);
xlabel('延时 k')
ylabel('R(k)')
legend('N=320')
axis([0,320,- 0.5,1]);

b2=b(1:320);
N=160;                                  % 选择的窗长
B=[];
for k=1:160;
sum=0;
for m=1:N;
sum=sum+ b2(m)* b2(m+ k- 1);
end
B(k)=sum;
end
for k=1:160
B1(k)=B(k)/B(1);                        % 归一化 B(k)
end
figure(1)
subplot(3,1,2)
plot(B1);
xlabel('延时 k')
```

```
ylabel('R(k)')
legend('N=160')
axis([0,320,- 0.5,1]);

b3=b(1:140);                                    % 选择的语音起始点
N=70;                                           % 选择的窗长
C=[];
for k=1:70;
sum=0;
for m=1:N;
sum=sum+ b3(m)* b3(m+ k- 1);
end
C(k)=sum;
end
for k=1:70
C1(k)=C(k)/C(1);                                % 归一化 C(k)
end
figure(1)
subplot(3,1,3)
plot(C1);
xlabel('延时 k')
ylabel('R(k)')
legend('N=70')
axis([0,320,- 0.5,1]);
```

因为求和上限是 $N-1$，与 k 无关，故当 k 增加时，$\hat{R}_n(k)$值不下降。与图 3.14 对应的修正自相关函数示于图 3.17 中。可以看到，自相关函数相关峰值下降很小。式(3.24)可以看做两个不同的有限长度段 $x(n+m)\hat{w}_1(m)$与 $x(n+m)\hat{w}_2(m)$的互相关函数。故 $\hat{R}_n(k)$有互相关函数的性质，而不具备自相关函数的性质，即 $\hat{R}_n(k)=\hat{R}_n(-k)$等，但这个 $\hat{R}_n(k)$的最近的第二个最大值点仍代表了基音周期的位置，而使 N 的长度压缩到最小，K 值可以做到大于 N 值。

计算短时自相关函数需要很大的运算量，有时为简化运算，常使用一种与自相关函数有相似作用的另一参量，即短时平均幅度差函数(AMDF)。

3.6.4 短时平均幅度差函数

对一个周期为 P 的周期信号 $x(n)$来说，在 $k=0,\pm P,\pm 2P,\cdots$时，$d(n)=x(n)-x(n-k)=0$，($k=0,\pm P,\pm 2P,\cdots$)。对于浊音语音，在基音周期的整数倍上，$d(n)$总是很小，但不是零，因此，可以定义短时平均幅度差函数 AMDF 为

$$r_n(k)=\sum_{m=-\infty}^{+\infty}|x(n+m)w_1(m)-x(n+m-k)w_2(m-k)| \tag{3.27}$$

显然，如果 $x(n)$具有周期 P，则当 $k=P,\pm 2P,\cdots$时，$r_n(k)$具有最小值。应该注意的是，取矩形窗是很合适的。如果 $w_1(n)$和 $w_2(n)$有同样的宽度，可得到类似于式(3.27)的幅度差函数；如果两个窗口长度不同，则将得到类似于修正自相关函数的函数。使用矩形窗时，短时

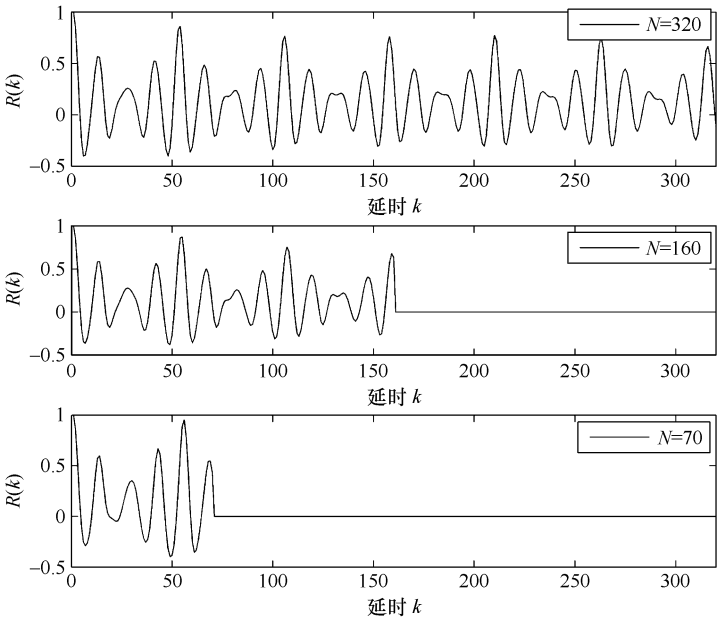


图 3.17　不同矩形窗长时的修正短时自相关函数

平均幅度差函数可写成

$$r_n(k)=\sum_{n=0}^{N-1}\mid x(n)-x(n+k)\mid,k=0,1,\cdots,N-1 \tag{3.28}$$

$r_n(k)$与$\hat{R}_n(k)$之间的关系为

$$r_n(k)\approx\sqrt{2}\beta(k)[\hat{R}_n(0)-\hat{R}_n(k)]^{1/2} \tag{3.29}$$

式中，$\beta(k)$对不同语音段可在 0.6～1.0 之间变化，但对于一个特定的语音段，它随 k 值的变化并不明显。

3.7　基于能量和过零率的语音端点检测

在复杂的应用环境下，从信号流中分辨出语音信号和非语音信号，是语音处理的一个基本问题。语音端点检测就是指从包含语音的一段信号中确定出语音的起始点和结束点。正确的端点检测对于语音识别和语音编码系统都有重要的意义，它可以使采集的数据真正是语音信号的数据，从而减少数据量和运算量并减少处理时间。

判别语音段的起始点和终止点的问题主要归结为区别语音和噪声的问题。如果能够保证系统的输入信噪比很高(即使最低电平的语音的能量也比噪声能量要高)，那么只要计算输入信号的短时能量就基本能够把语音段和噪声背景区别开来。但是，在实际应用中很难保证这么高的信噪比，仅仅根据能量来判断是比较粗糙的。因此，还需进一步利用短时平均过零率进行判断，因为清音和噪声的短时平均过零率比背景噪声的平均过零率要高出好几倍。本节介绍基于能量和过零率的语音端点检测方法——两级判决法及程序实现。

两级判决法采用双门限比较法，可以用图 3.18 来说明。

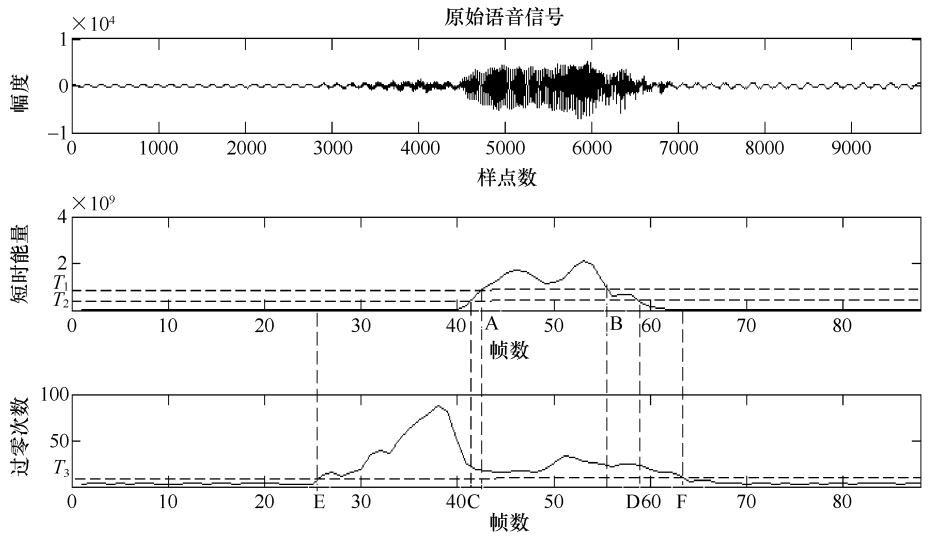


图 3.18　利用能量和过零率进行语音端点检测的两级判决法示意图

1. 第一级判决

① 先根据语音短时能量的轮廓选取一个较高的门限 T_1，进行一次粗判：语音起止点位于该门限与短时能量包络交点所对应的时间间隔之外（即 AB 段之外）。

② 根据背景噪声的平均能量确定一个较低的门限 T_2，并从 A 点往左、从 B 点往右搜索，分别找到短时能量包络与门限 T_2 相交的两个点 C 和 D，于是 CD 段就是用双门限方法根据短时能量所判定的语音段。

2. 第二级判决

以短时平均过零率为标准，从 C 点往左和从 D 点往右搜索，找到短时平均过零率低于某个门限 T_3 的两点 E 和 F，这便是语音段的起止点。门限 T_3 是由背景噪声的平均过零率所确定的。

这里要注意，门限 T_2，T_3 都是由背景噪声特性确定的，因此，在进行起止点判决前，通常都要采集若干帧背景噪声并计算其平均短时能量和平均过零率，作为选择 T_2 和 T_3 的依据。当然，T_1，T_2，T_3，三个门限值的确定还应当通过多次实验。

基于 MATLAB 程序实现能量与过零率的端点检测算法步骤如下：

① 语音信号 $x(n)$ 进行分帧处理，每一帧记为 $s_i(n)$，$n=1,2,\cdots,N$，n 为离散语音信号时间序列，N 为帧长，i 表示帧数。

② 计算每一帧语音的短时能量，得到语音的短时帧能量：$E_i = \sum_{n=1}^{N} s_i^2(n)$。

③ 计算每一帧语音的过零率，得到短时帧过零率：$Z_i = \sum_{n=1}^{N} |\,\mathrm{sgn}[s_i(n)] - \mathrm{sgn}[s_i(n-1)]\,|$。

其中

$$\mathrm{sgn}[s_i(n)] = \begin{cases} 1, & s_i(n) \geqslant 0 \\ 0, & s_i(n) < 0 \end{cases}$$

④考察语音的平均能量设置一个较高的门限 T_1，用以确定语音开始，然后再根据背景噪声的平均能量确定一个稍低的门限 T_2，用以确定第一级中的语音结束点。$T_2 = \alpha_1 E_N$，E_N 为噪

声段能量的平均值。完成第一级判决。第二级判决同样根据背景噪声的平均过零率 Z_N，设置一个门限 T_3，用于判断语音前端的清音和后端的尾音。α_1，β_1 为经过大量实验得到的经验值。

由于 MATLAB 实现的程序较长，这里从略。

3.8 基音周期估值

基音周期是表征语音信号本质特征的参数，属于语音分析的范畴，只有准确分析并且提取出语音信号的特征参数，才能够利用这些参数进行语音编码、语音合成和语音识别等处理。语音编码的压缩率高低、语音合成的音质好坏及语音识别率的高低，也都依赖于对语音信号分析的准确性和精确性，因此基音周期估值在语音信号处理应用中具有十分重要的作用。语音信号基音周期估值的方法很多，本节介绍最基本的两种方法：基于短时自相关法的基音周期估值和基于短时平均幅度差函数法的基音周期估值。

3.8.1 基于短时自相关法的基音周期估值

前文介绍过自相关函数的性质，如果 $x(n)$ 是一个周期为 P 的信号，则其自相关函数也是周期为 P 的信号，且在信号周期的整数倍处，自相关函数取最大值。语音的浊音信号具有准周期性，其自相关函数在基音周期的整数倍处取最大值。计算两相邻最大峰值间的距离，就可以估计出基音周期。观察浊音信号的自相关函数图，其中真正反映基音周期的只是其中少数几个峰，而其余大多数峰都是由于声道的共振特性引起的。因此为了突出反映基音周期的信息，同时压缩其他无关信息，减小运算量，有必要对语音信号进行适当预处理后再进行自相关计算以获得基音周期。

第一种方法是先对语音信号进行低通滤波，再进行自相关计算。因为语音信号包含十分丰富的谐波分量，基音频率的范围分布在 50～500Hz，即使女高音升 C 调最高也不会超过 1kHz，所以采用 1kHz 的低通滤波器先对语音信号进行滤波，保留基音频率；再用 2kHz 采样频率进行采样；最后用 2～20ms 的滞后时间计算短时自相关，帧长取 10～20ms，即可估计出基音周期。

第二种方法是先对语音信号进行中心削波处理，再进行自相关计算。常用的有两种削波函数，下面分别介绍。

1. 中心削波

中心削波函数如式(3.30)所示，其对应波形如图 3.19 所示。

$$f(x)=\begin{cases}x-x_L & (x>x_L)\\ 0 & (-x_L\leqslant x\leqslant x_L)\\ x+x_L & (x<-x_L)\end{cases} \tag{3.30}$$

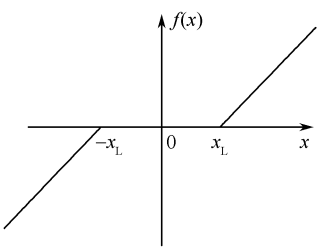


图 3.19 中心削波函数

一般削波电平 x_L 取本帧语音最大幅度的 60%～70%。将削波后的序列 $f(x)$ 用短时自相关函数估计基音周期，在基音周期位置的峰值更加尖锐，可以有效减少倍频或半频错误。图 3.20 和图 3.21 分别给出了削波前后语音信号对比图及修正自相关对比图。

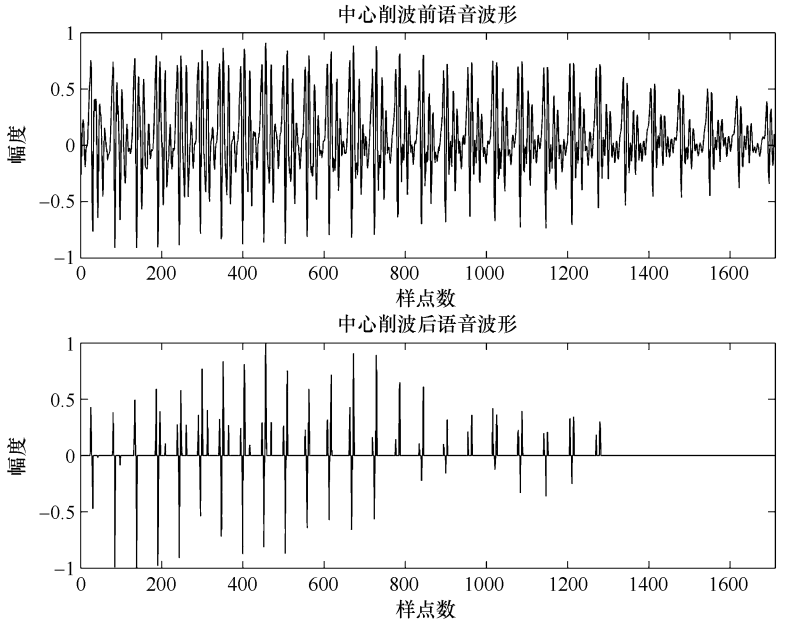


图 3.20　中心削波前后语音信号对比图

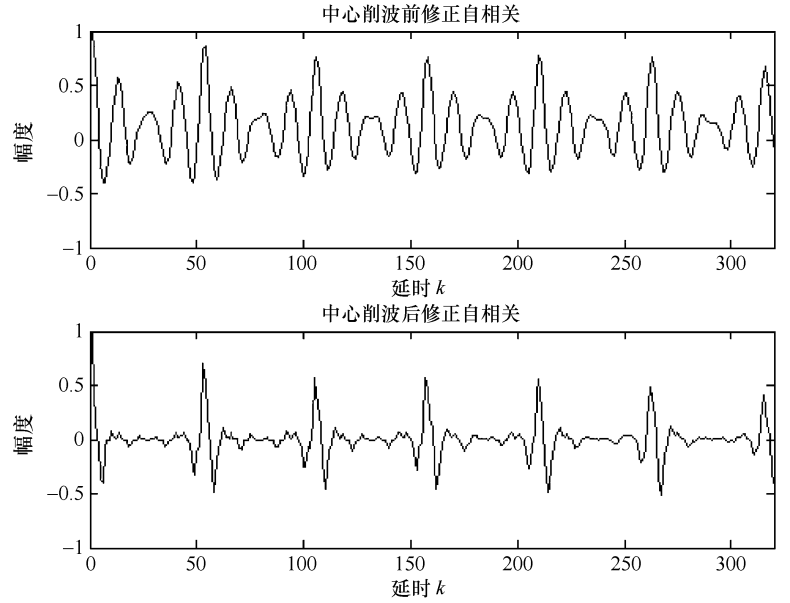


图 3.21　中心削波前后修正自相关对比图

【程序 3.10】zhongxinxuebo.m

```
% 本程序运行结果为中心削波前后的语音波形,以及削波前后的自相关波形
% 读入数据 采样 fs=8kHz 采样位数 16bit 长度 320 样点
fid=fopen('voice.txt','rt');                    % 打开语音文件
[a,count]=fscanf(fid,'% f',[1,inf]);            % 读语音文件
L=length(a);                                    % 测定语音的长度
m=max(a);
for i=1:L
  a(i)=a(i)/m;                                  % 数据归一化
```

```
end

% 找到归一化以后数据的最大值和最小值
m=max(a);                                   % 找到最大的正值
n=min(a);                                   % 找到最小的负值
% 为保证幅度值与横坐标轴对称,采用计算公式是 n+ (m- n)/2,合并为(m+ n)/2
ht=(m+ n)/2;
for i=1:L;                                  % 数据中心下移,保持和横坐标轴对称
  a(i)=a(i)- ht;
end
figure(1);                                  % 画第一幅图
subplot(2,1,1);                             % 第一个子图
plot(a,'k');
axis([0,1711,- 1,1]);                       % 确定横、纵坐标的范围
title('中心削波前语音波形');                % 图标题
xlabel('样点数');                           % 横坐标
ylabel('幅度');                             % 纵坐标

coeff=0.7;                                  % 中心削波函数系数取 0.7
th0=max(a)* coeff;                          % 求中心削波函数门限(threshold)
for k=1:L ;                                 % 中心削波
    if a(k)> =th0
        a(k)=a(k)- th0;
    elseif a(k)< =(- th0);
        a(k)=a(k)+ th0;
    else
        a(k)=0;
    end
end
m=max(a);
for i=1:L;                                  % 中心削波函数幅度的归一化
  a(i)=a(i)/m;
end
subplot(2,1,2);                             % 第二个子图
plot(a,'k');
axis([0,1711,- 1,1]);                       % 确定横、纵坐标的范围
title('中心削波后语音波形');                % 图标题
xlabel('样点数');                           % 横坐标
ylabel('幅度');                             % 纵坐标
fclose(fid);                                % 关闭文件

% 没有经过中心削波的修正自相关计算
fid=fopen('voice.txt','rt');
[b,count]=fscanf(fid,'% f',[1,inf]);
fclose(fid);
```

```
N=320;                              % 选择的窗长
A=[];
for k=1:320;                        % 选择延迟长度
sum=0;
for m=1:N;
  sum=sum+ b(m)* b(m+ k- 1);        % 计算自相关
end
A(k)=sum;
end
for k=1:320
    B(k)=A(k)/A(1);                 % 自相关归一化
end
figure(2);                          % 画第二幅图
subplot(2,1,1);                     % 第一个子图
plot(B,'k');
title('中心削波前修正自相关');      % 图标题
xlabel('延时 k');                   % 横坐标
ylabel('幅度');                     % 纵坐标
axis([0,320,- 1,1]);

% 中心削波函数和修正的自相关方法结合
N=320;                              % 选择的窗长
A=[];
for k=1:320;                        % 选择延迟长度
sum=0;
for m=1:N;
  sum=sum+ a(m)* a(m+ k- 1);        % 对削波后的函数计算自相关
end
A(k)=sum;
end
for k=1:320
    C(k)=A(k)/A(1);                 % 自相关归一化
end

subplot(2,1,2);                     % 第二个子图
plot(C,'k');
title('中心削波后修正自相关');      % 图标题
xlabel('延时 k');                   % 横坐标
ylabel('幅度');                     % 纵坐标
axis([0,320,- 1,1]);
```

2. 三电平削波

为了克服短时自相关函数计算量大的问题，在中心削波法的基础上，还可以采用三电平削波法，削波函数如式(3.31)所示，其波形表示如图 3.22 所示。

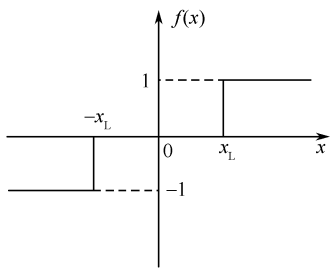


图 3.22　三电平削波函数

$$f(x)=\begin{cases}1 & x>x_{\mathrm{L}}\\ 0 & -x_{\mathrm{L}}\leqslant x\leqslant x_{\mathrm{L}}\\ -1 & x<-x_{\mathrm{L}}\end{cases} \tag{3.31}$$

经削波后的取样值仅有三种可能情况，即+1，0，−1。显然，这种信号的短时自相关函数的计算实际上是不需要乘法运算的，这就大大节省了计算时间。图 3.23 和图 3.24 分别画出了削波前后语音信号对比图及修正自相关对比图。由于实现程序与中心削波程序相似，这里不再给出，大家可以自己编制。

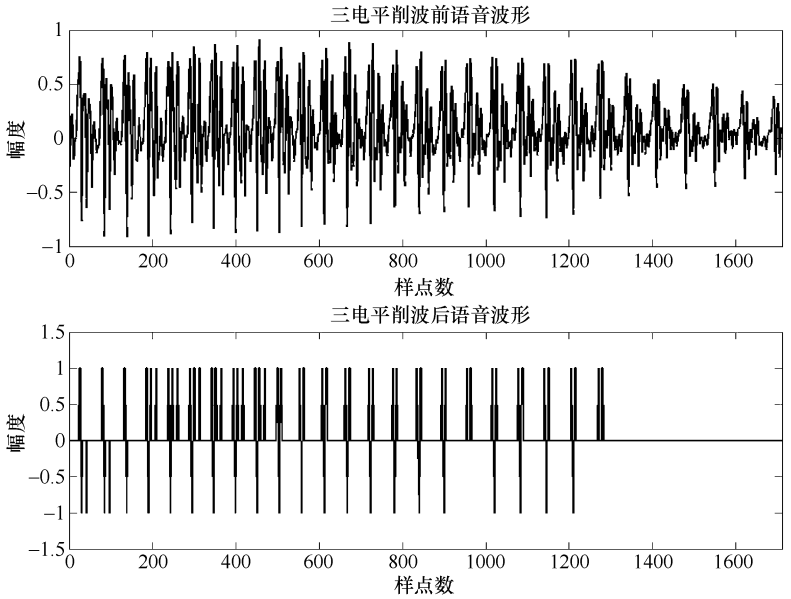


图 3.23　三电平削波前后语音信号对比图

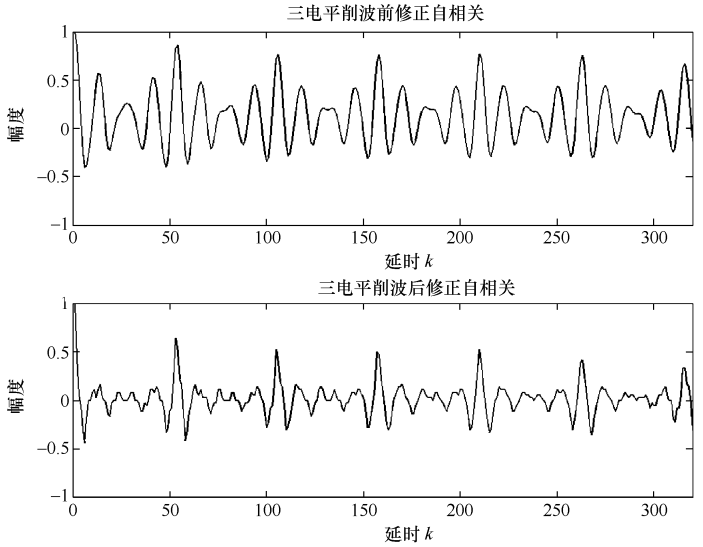


图 3.24　三电平削波前后修正自相关对比图

3.8.2 基于短时平均幅度差函数 AMDF 法的基音周期估值

根据 3.6.4 节关于短时平均幅度差函数的介绍，可以知道：如果信号 $x(n)$是标准的周期信号，则相距为周期的整数倍的样点上的幅度值是相等的，二者差值为零。对于浊音语音，在基音周期的整数倍上，这个差值不是零，但总是很小，因此，我们可以通过计算短时平均幅度差函数中两相邻谷值间的距离来进行基音周期估值。这里使用修正的短时平均幅度差函数并加矩形窗，得

$$r_n(k)=\sum_{n=0}^{N-1}|x(n)-x(n+k)|,k=0,1,\cdots,N-1 \tag{3.32}$$

显然，如果 $x(n)$具有周期 P，则当 $k=\pm P,\pm 2P,\cdots$时，$r_n(k)$具有最小值。图 3.25 给出了一段浊音信号及其 AMDF 函数的波形。与短时自相关函数的不同是：自相关函数进行基音周期估计时寻找的是最大峰值点的位置，而 AMDF 寻找的是它的最小谷值点的位置。由于清音没有周期性，所以它的自相关函数和平均幅度差函数均不具有准周期性的峰值或谷值。实现图 3.25 的程序如下所示。

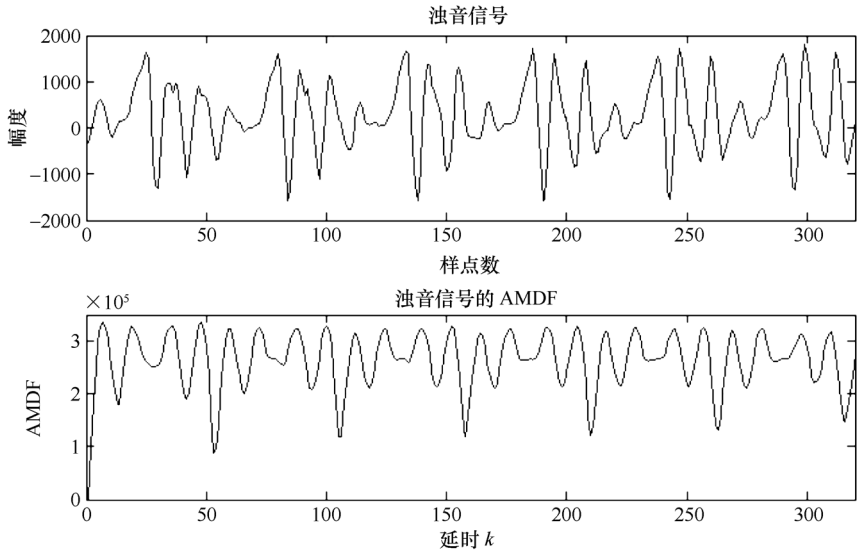


图 3.25 一段浊音信号及其 AMDF 函数

【程序 3.11】AMDF.m

```
fid=fopen('voice.txt','rt')
[b,count]=fscanf(fid,'% f',[1,inf]);
fclose(fid);

b1=b(1:640);
N=320;                              % 选择的窗长
A=[];
for k=1:320;
  sum=0;
  for m=1:N;
  sum=sum+ abs(b1(m)- b1(m+ k- 1));
  end
```

```
  A(k)=sum;
end

s=b(1:320)
figure(1)
subplot(2,1,1)
plot(s);
xlabel('样点')
ylabel('幅度')
axis([0,320,- 2* 10^3,2* 10^3])
subplot(2,1,2)
plot(A);
xlabel('延时 k')
ylabel('AMDF')
axis([0,320,0,3.5* 10^5]);
```

3.8.3 基音周期估值的后处理

语音信号中的浊音信号的周期性从波形上观察可以看得很明显，但是其形状比较复杂，这使得基音检测算法很难做到处处准确可靠。在提取基音的过程中，无论采用哪种方法提取的基音频率轨迹与真实的基音频率轨迹都不可能完全吻合。实际情况是大部分段落吻合，而在一些局部段落和区域中有一个或几个基音频率估计值偏离，甚至远离正常轨迹，通常是偏离到正常值的 2 倍或 1/2 处，即实际基音频率的倍频或分频处，称这种偏离点为基音轨迹的“野点”。

为了去除这些“野点”，对求得的基音轨迹进行平滑后处理是非常必要的。语音信号的基频通常是连续缓慢变化的，因此，用某种平滑技术来纠正这些“野点”是可以的。常用的平滑技术主要有：中值滤波平滑处理、线性平滑、动态规划平滑处理。

1. 中值平滑处理

中值平滑处理的基本原理是：设 $x(n)$为输入信号，$y(n)$为中值滤波器的输出，采用一滑动窗，则 n_0处的输出值 $y(n_0)$就是将窗的中心移到 n_0处时窗内输入样点的中值。即在 n_0点的左右各取 L 个样点。连同被平滑点共同构成一组信号采样值[共$(2L+1)$个样值]，然后将这$(2L+1)$个样值按大小次序排成一队，取此队列中的中间者作为平滑器的输出。L 值一般取为 1 或 2，即中值平滑的“窗口”一般包括 3～5 个样值，称为 3 点或 5 点中值平滑。中值平滑的优点是既可以有效地去除少量的“野点”，又不会破坏基音周期轨迹中两个平滑段之间的阶跃性变化。

2. 线性平滑处理

线性平滑是用滑动窗进行线性滤波处理，即

$$y(n)=\sum_{m=-L}^{L}x(n-m)w(m) \tag{3.33}$$

其中，$\{w(m),m=-L,-L+1,\cdots,0,1,2,\cdots,L\}$为 $2L+1$ 点平滑窗，满足

$$\sum_{m=-L}^{L}w(m)=1 \tag{3.34}$$

例如三点窗的权值可取为$\{0.25,0.5,0.25\}$。线性平滑在纠正输入信号中不平滑样点值的同

时，也使附近各样点的值做了修改。所以窗的长度加大虽然可以增加平滑的效果，但是同时也可能导致两个平滑段之间阶跃的模糊程度加重。将以上两种平滑技术结合起来使用可以克服各自的不足。

3. 组合平滑处理

为了改善平滑的效果可以将两个中值平滑串接，图 3.26(a)所示是将一个 5 点中值平滑和一个 3 点中值平滑串接。另一种方法是将中值平滑和线性平滑组合，如图 3.26(b)所示。为了使平滑的基音轨迹更为贴近，还可以采用二次平滑的算法。设所要平滑的信号为 $T_P(n)$，经过一次组合得到的信号为 $\tau_P(n)$。那么首先应求出两者的差值信号 $\Delta T_P(n)=T_P(n)-\tau_P(n)$，再对 $\Delta T_P(n)$进行组合平滑，得到 $\Delta\tau_P(n)$，则输出等于 $\tau_P(n)+\Delta\tau_P(n)$，就可以得到更好的基音周期估计轨迹。全部算法的框图如图 3.26(c)所示。由于中值平滑和线性平滑都会引入延时，所以在实现上述方案时应考虑到它的影响。图 3.26(d)是一个采用补偿延时的可实现二次平滑方案。其中的延时大小可由中值平滑的点数和线性平滑的点数来决定。例如，一个 5 点的中值平滑引入 2 点延时，一个 3 点平滑引入 1 点延时，那么采用此两者完成组合平滑时，补偿延时的点数应等于 3。

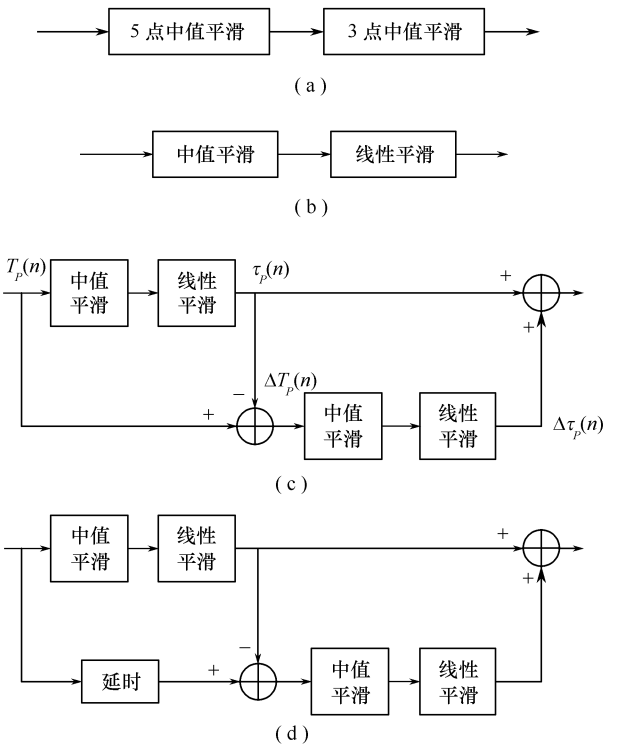


图 3.26　各种组合平滑算法

3.8.4　基音周期估值后处理的 MATLAB 实现

本实验所用的语音样本是用 Cooledit 在普通室内环境下录制的女声“我到北京去”，采样频率为 8kHz，单声道，将语音信号分为若干帧，每帧长 220 个样点，相邻帧交叠 110 个样点，采用基于能量的基音周期检测算法求出基音周期，并将原始基音周期保存为“zhouqi. txt”文件，用下面程序对原始基音周期进行平滑处理。

【程序 3.12】

```
fid=fopen('zhouqi.txt','rt');          % 读入语音文件
zhouqi=fscanf(fid,'% f');
fclose(fid);
zhouqi0=medfilt1(zhouqi,5);            % 五点中值平滑
zhouqi1=medfilt1(zhouqi0,3);           % 三点中值平滑,zhouqi1 为五点中值平滑和三
                                         点中值平滑组合
zhouqi2=linsmooth(zhouqi0,5);          % 五点线性平滑,zhouqi2 为五点中值平滑和五
                                         点线性平滑组合
w=[];
w=zhouqi;
w1=w- zhouqi2;
w1=medfilt1(w1,5);                     % 五点中值平滑
w1=linsmooth(w1,5);                    % 五点线性平滑
zhouqi3=w1+ zhouqi2;                   % 二次平滑算法

v=[];
v(1)=0;v(2)=0;v(3)=0;v(4)=0;           % 延时 4 个样点
for i=1:(length(zhouqi)- 4)
    v(i+ 4)=zhouqi(i);
end
v=v(:);
v1=v- zhouqi2;
v1=medfilt1(v1,5);                     % 五点中值平滑
v1=linsmooth(v1,5);                    % 五点线性平滑
zhouqi4=v1+ zhouqi2;                   % 加延时的二次平滑算法

figure(1)
subplot(511)
plot(zhouqi);
xlabel('帧数')
ylabel('样点数')
axis([0,360,0,150])
title('原始基音周期轨迹')

subplot(512),plot(zhouqi2);
xlabel('帧数')
ylabel('样点数')
axis([0,360,0,150])
title('五点中值平滑和三点中值平滑组合')

subplot(513),plot(zhouqi2);
xlabel('帧数')
ylabel('样点数')
```

```
axis([0,360,0,150])
title('五点中值平滑和五点线性平滑组合')

subplot(514),plot(zhouqi3);
xlabel('帧数')
ylabel('样点数')
axis([0,360,0,150])
title('二次平滑算法')

subplot(515),plot(zhouqi4);
xlabel('帧数')
ylabel('样点数')
axis([0,360,0,150])
title('加延时的二次平滑算法')
```

其中,linsmooth()函数的 MATLAB 程序如下:

```
function [y] = linsmooth(x,n,wintype)
% linsmooth(x,wintype,n) : linear smoothing
% x: 输入
% n: 窗长
% wintype: 窗类型,默认为 'hann'
if nargin< 3
   wintype='hann';
end
if nargin< 2
   n=3;
end
win=hann(n);
win=win/sum(win);                    % 归一化
[r,c]=size(x);
if min(r,c)~=1
   error('sorry, no matrix here!:(')
end

if r==1                              % 行向量
    len=c;
else
   len=r;
   x=x.';
end
y=zeros(len,1);
if mod(n,2)==0
   l=n/2;
   x = [ones(1,l)* x(1) x ones(1,l)* x(len)]';
else
```

```
    l=(n-1)/2;
    x = [ones(1,l)*x(1) x ones(1,l+1)*x(len)]';
end

for k=1:len
    y(k) = win'*x(k:k+n-1);
end
```

程序运行结果如图 3.27 所示，可以看出，组合平滑算法对原始基音周期的“野点”有很好的平滑作用，二次平滑算法在对语音“我到北京去”的平滑作用上，与组合平滑算法相差无几，都很好地实现了对原始语音进行平滑。理论上加延时的二次平滑算法的平滑效果应优于二次平滑算法，但在该实验中效果不佳，可能原因是原始基音周期已经趋于平滑，加延时反而造成基音周期的不准确。

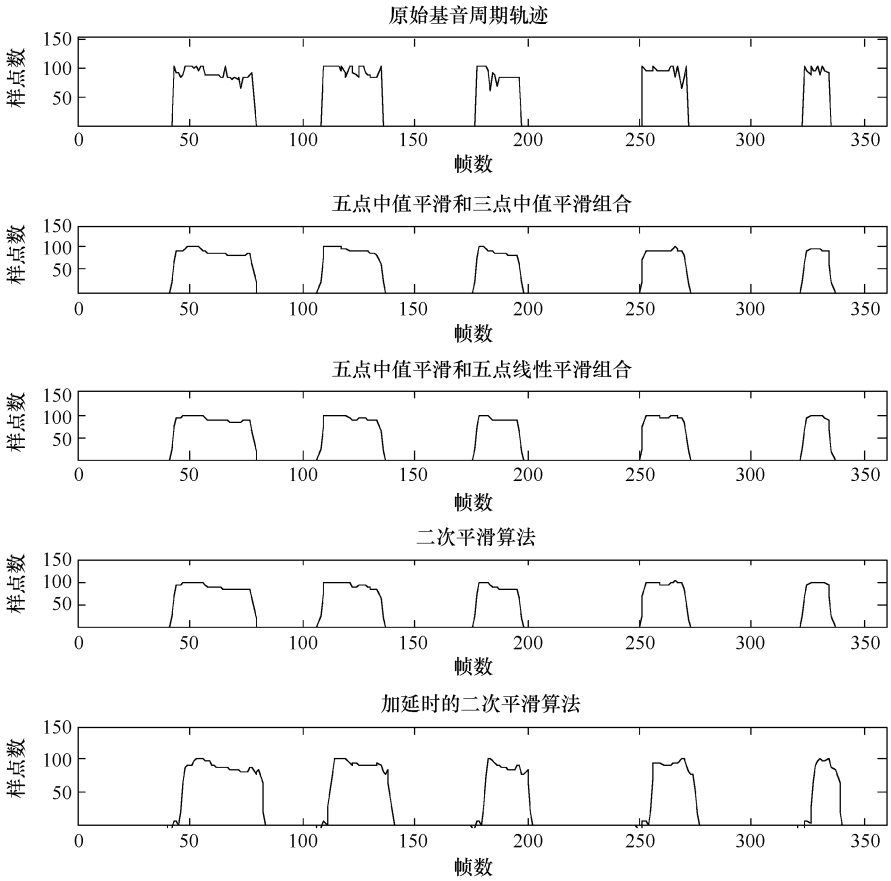


图 3.27　各种组合平滑算法运行结果

第 4 章　语音信号短时频域分析

4.1　概　　述

傅里叶分析是分析线性系统和平稳信号稳态特性的强有力工具，它在许多工程领域得到了广泛应用。它理论完善，且有快速算法，在语音信号处理领域也是一个重要工具。

语音信号本质上是非平稳信号，其非平稳特性是由发声器官的物理运动过程产生的。发声器官的运动由于存在惯性，所以可以假设语音信号在 10～30ms 这样短的时间段内是平稳的，这是短时分帧处理的基础，也是短时傅里叶分析的基础。短时傅里叶分析就是在基于短时平稳的假设下，用稳态分析方法处理非平稳信号的一种方法。

根据语音信号的二元激励模型，语音被看做一个受准周期脉冲或随机噪声源激励的线性系统的输出。输出频谱是声道系统的频率响应与激励源频谱的乘积，一般标准的傅里叶变换适用于周期及平稳随机信号的表示，但不能直接用于语音信号。因为语音信号可被看做短时平稳信号，所以可采用短时傅里叶分析。某一帧的短时傅里叶变换的定义式如下：

$$X_n(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{m=-\infty}^{+\infty} x(m)w(n-m)\mathrm{e}^{-\mathrm{j}\omega m} \tag{4.1}$$

式中，$w(n-m)$是窗函数。不同的窗函数，可得到不同的傅里叶变换的结果。在式中，短时傅里叶变换有两个变量，即离散时间 n 及连续频率 ω，若令 $\omega=2\pi k/N$，则可得到离散的短时傅里叶变换如下：

$$X_n(\mathrm{e}^{\mathrm{j}\frac{2\pi k}{N}}) = X_n(k) = \sum_{m=-\infty}^{+\infty} x(m)w(n-m)\mathrm{e}^{-\mathrm{j}\frac{2\pi km}{N}}, \quad 0 \leqslant k \leqslant N-1 \tag{4.2}$$

它实际上就是 $X_n(\mathrm{e}^{\mathrm{j}\omega})$的频率的抽样。由式(4.1)或式(4.2)可以看出：当 n 固定时，它们就是序列$[w(n-m)x(m)]$$(-\infty\leqslant m\leqslant+\infty)$的傅里叶变换或离散傅里叶变换；当 ω 或 k 固定时，它们是一个卷积，这相当于滤波器的运算。因此，语音信号的短时频域分析可以解释为傅里叶变换或滤波器。

4.2　傅里叶变换的解释

将式(4.1)写为

$$X_n(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{m=-\infty}^{+\infty} [x(m)w(n-m)]\mathrm{e}^{-\mathrm{j}\omega m} \tag{4.3}$$

时变傅里叶变换是时间 n 的函数，当 n 变化时，窗 $w(n-m)$沿着 $x(m)$滑动，图 4.1 画出了这种情况，它表明了在几个不同的 n 值上 $x(m)$及 $w(n-m)$与 m 的函数关系。

因为 $w(n-m)$为有限宽度窗，故 $x(m)w(n-m)$在所有 n 上绝对可和，因而时变傅里叶变换必定存在。另外，时变傅里叶变换也是 ω 的周期函数，且周期为 2π。当 n 固定时，时变傅里

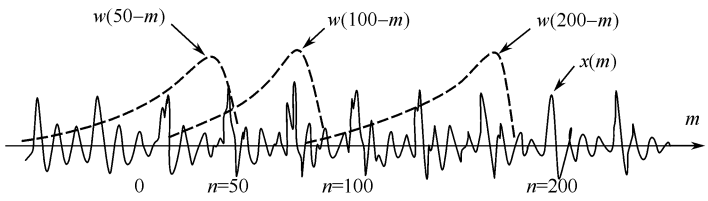


图 4.1　$x(m)$及 $w(n-m)$与 m 的函数关系

叶变换的特性与标准傅里叶变换相同，故可写出傅里叶逆变换公式为

$$w(n-m)x(m)=\frac{1}{2\pi}\int_{-\pi}^{\pi}X_n(\mathrm{e}^{\mathrm{j}\omega})\mathrm{e}^{\mathrm{j}\omega m}\,\mathrm{d}\omega \tag{4.4}$$

令 $m=n$，则

$$x(n)=\frac{1}{2\pi w(0)}\int_{-\pi}^{\pi}X_n(\mathrm{e}^{\mathrm{j}\omega})\mathrm{e}^{\mathrm{j}\omega n}\,\mathrm{d}\omega \tag{4.5}$$

从上式可以看出，只有当 $w(0)\neq 0$ 时，$x(n)$才能从 $X_n(\mathrm{e}^{\mathrm{j}\omega})$求出。

此外，由功率谱定义，可以写出短时功率谱与短时傅里叶变换的关系：

$$S_n(\mathrm{e}^{\mathrm{j}\omega})=X_n(\mathrm{e}^{\mathrm{j}\omega})X_n(\mathrm{e}^{\mathrm{j}\omega})=|X_n(\mathrm{e}^{\mathrm{j}\omega})|^2 \tag{4.6}$$

功率谱 $S_n(\mathrm{e}^{\mathrm{j}\omega})$是自相关函数

$$R_n(k)=\sum_{m=-\infty}^{+\infty}x(m)w(n-m)x(m+k)w(n-k-m) \tag{4.7}$$

的傅里叶变换。

下面讨论窗函数的作用。

对于 $w(n-m)$窗来说，它除了具有选出 $x(m)$序列中被分析部分的作用外，它的形状对时变傅里叶变换的特性也有重要作用，从标准傅里叶变换可以方便地解释这种作用。如果 $X_n(\mathrm{e}^{\mathrm{j}\omega n})$被看成是 $w(n-m)x(m)$序列的标准傅里叶变换，同时假设 $x(m)$及 $w(m)$的标准傅里叶变换存在，即

$$X(\mathrm{e}^{\mathrm{j}\omega})=\sum_{m=-\infty}^{+\infty}x(m)\mathrm{e}^{-\mathrm{j}\omega m} \tag{4.8}$$

$$W(\mathrm{e}^{\mathrm{j}\omega})=\sum_{m=-\infty}^{+\infty}w(m)\mathrm{e}^{-\mathrm{j}\omega m} \tag{4.9}$$

当 n 固定时，序列 $w(n-m)$的傅里叶变换为

$$\sum_{m=-\infty}^{+\infty}w(n-m)\mathrm{e}^{-\mathrm{j}\omega m}=W(\mathrm{e}^{-\mathrm{j}\omega})\mathrm{e}^{-\mathrm{j}\omega n} \tag{4.10}$$

根据卷积定理，两相乘序列的傅里叶变换等于各自傅里叶变换的卷积，因此，$w(n-m)x(m)$序列的标准傅里叶变换 $X_n(\mathrm{e}^{\mathrm{j}\omega n})$为

$$X_n(\mathrm{e}^{\mathrm{j}\omega})=[W(\mathrm{e}^{-\mathrm{j}\omega})\cdot\mathrm{e}^{-\mathrm{j}\omega n}]*[X(\mathrm{e}^{\mathrm{j}\omega})] \tag{4.11}$$

因为式(4.11)右边两个卷积项都是 ω 的周期为 2π 的连续周期函数，所以上式可写成卷积积分的形式

$$X_n(\mathrm{e}^{\mathrm{j}\omega})=\frac{1}{2\pi}\int_{-\pi}^{\pi}W(\mathrm{e}^{-\mathrm{j}\theta})\mathrm{e}^{-\mathrm{j}\theta n}\cdot X(\mathrm{e}^{\mathrm{j}(\omega-\theta)})\,\mathrm{d}\theta \tag{4.12}$$

将 θ 改换为 $-\theta$ 后，可以写成

$$X_n(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\theta})e^{j\theta n} \cdot X(e^{j(\omega+\theta)})d\theta \tag{4.13}$$

式(4.13)表示在 $-\infty<m<\infty$ 区间内，$x(m)$ 序列的傅里叶变换与平移窗序列 $w(n-m)$ 的傅里叶变换的卷积。从式(4.13)中可以看出，为了使 $X_n(e^{j\omega})$ 能够充分地表现 $X(e^{j\omega})$ 的特性，要求 $W(e^{j\theta})$ 对于 $X(e^{j\omega})$ 来说必须是一个冲激脉冲。

选择的窗函数和窗宽的不同，对短时傅里叶谱的影响是不同的。

图 4.2 为加不同窗函数时的清浊音波形及频谱图。语音信号采样率为 8kHz，窗长取 256。可以看出在矩形窗和汉明窗两种窗函数下，短时频谱图都有两种变化：由周期性激励引起的快变化，反映了基音频率的各次谐波；由声道的共振特性引起的慢变化，反映了各共振峰的频率和带宽。还可以看出两个频谱图之间存在明显的差别。采用矩形窗时，基音谐波的各个峰都比较尖锐，且整个频谱图显得比较破碎(类似于噪声)，这是因为矩形窗的主瓣较窄，具有较高的频率分辨率，但它也具有较高的旁瓣，因而使基音的相邻谐波之间的干扰比较严重。在相邻谐波间隔内有时叠加，有时抵消，出现了一种随机变化的现象。相邻谐波之间的这种严重“泄露”的现象，抵消了矩形窗主瓣窄的优点，因此，在语音短时频谱分析中极少采用矩形窗。当加汉明窗时，得到的短时频谱要平滑得多，因而在语音分析中汉明窗用得比较普遍。其 MATLAB 程序如下。

【程序 4.1】 qingzhuoyinpinpu.m

```
fid=fopen('voice2.txt','rt');          % 打开文件
y=fscanf(fid,'% f');                   % 读数据
e=fra(256,128,y);                      % 对 y 分帧,帧长 256,帧移 128
ee=e(10,:);                            % 选取第 10 帧
subplot(421)                           % 画第 1 个子图
ee1=ee/max(ee);                        % 幅值归一化
plot(ee1)                              % 画波形
xlabel('样点数')                       % 横坐标名称
ylabel('幅度')                         % 纵坐标名称
title ('原始语音')                     % 文字标注
axis([0,256,- 1.5,1.5])                % 限定横纵坐标范围

% 矩形窗傅里叶变换
r=fft(ee,1024);                        % 对信号 ee 进行 1024 点傅里叶变换
r1=abs(r);                             % 对 r 取绝对值 r1 表示频谱的幅度值
r1=r1/max(r1);                         % 幅值归一化
yuanlai=20* log10(r1);                 % 对归一化幅值取对数
signal(1:256)=yuanlai(1:256);          % 取 256 个点,目的是画图的时候,维数一致
pinlv=(0:1:255)* 8000/512;             % 点和频率的对应关系
subplot(425)                           % 画第 5 个子图
plot(pinlv,signal);                    % 画幅值特性图
xlabel('频率/Hz')                      % 横坐标名称
ylabel('对数幅度/dB')                  % 纵坐标名称
title ('加矩形窗时语音谱')             % 文字标注
```

```
axis([0,4000,- 80,15])                    % 限定横纵坐标范围

% 加汉明窗
f=ee'.* hamming(length(ee));              % 对选取的语音信号加汉明窗
f1=f/max(f);                              % 对加窗后的语音信号的幅值归一化
subplot(423)                              % 画第 3 个子图
plot(f1)                                  % 画波形
axis([0,256,- 1.5,1.5])                   % 限定横纵坐标范围
xlabel('样点数')                          % 横坐标名称
ylabel('幅度')                            % 纵坐标名称
title ('窗选语音')                        % 文字标注

% 加汉明窗傅里叶变换
r=fft(f,1024);                            % 对信号 ee 进行 1024 点傅里叶变换
r1=abs(r);                                % 对 r 取绝对值 r1 表示频谱的幅度值
r1=r1/max(r1);                            % 幅值归一化
yuanlai=20* log10(r1);                    % 对归一化幅值取对数
signal(1:256)=yuanlai(1:256);             % 取 256 个点,目的是画图的时候,维数一致
pinlv=(0:1:255)* 8000/512;                % 点和频率的对应关系
subplot(427)                              % 画第 7 个子图
plot(pinlv,signal);                       % 画幅值特性图
xlabel('频率/Hz')                         % 横坐标名称
ylabel('对数幅度/dB')                     % 纵坐标名称
title ('加汉明窗时语音谱')                % 文字标注
axis([0,4000,- 80,15])                    % 限定横纵坐标范围

% 清音的波形和短时频谱图(窗长 256)
fid=fopen('qingyin1.txt','rt');           % 打开文件
y=fscanf(fid,'% f');                      % 读数据
e=fra(256,128,y);                         % 对 y 分帧,帧长 256,帧移 128
ee=e(2,:);                                % 选取第 2 帧
subplot(422)                              % 画第 2 个子图
ee1=ee/max(ee);                           % 幅值归一化
plot(ee1)                                 % 画波形
xlabel('样点数')                          % 横坐标名称
ylabel('幅度')                            % 纵坐标名称
title ('原始语音')                        % 文字标注
axis([0,256,- 1.5,1.5])                   % 限定横纵坐标范围

% 矩形窗傅里叶变换
r=fft(ee,1024);                           % 对信号 ee 进行 1024 点傅里叶变换
r1=abs(r);                                % 对 r 取绝对值 r1 表示频谱的幅度值
r1=r1/max(r1);                            % 幅值归一化
yuanlai=20* log10(r1);                    % 对归一化幅值取对数
```

```
signal(1:256)=yuanlai(1:256);          % 取 256 个点,目的是画图的时候,维数一致
pinlv=(0:1:255)* 8000/512;             % 点和频率的对应关系
subplot(426)                           % 画第 6 个子图
plot(pinlv,signal);                    % 画幅值特性图
xlabel('频率/Hz')                      % 横坐标名称
ylabel('对数幅度/dB')                  % 纵坐标名称
title('加矩形窗时语音谱')              % 文字标注
axis([0,4000,- 80,1])                  % 限定横纵坐标范围

% 加汉明窗
f=ee'.* hamming(length(ee));           % 对选取的语音信号加汉明窗
f1=f/max(f);                           % 对加窗后的语音信号的幅值归一化
subplot(424)                           % 画第 4 个子图
plot(f1)                               % 画波形
axis([0,256,- 1.5,1.5])                % 限定横纵坐标范围
xlabel('样点数')                       % 横坐标名称
ylabel('幅度')                         % 纵坐标名称
title ('窗选语音')                     % 文字标注

% 加汉明傅里叶变换
r=fft(f,1024);                         % 对信号 ee 进行 1024 点傅里叶变换
r1=abs(r);                             % 对 r 取绝对值 r1 表示频谱的幅度值
r1=r1/max(r1);                         % 幅值归一化
yuanlai=20* log10(r1);                 % 对归一化幅值取对数
signal(1:256)=yuanlai(1:256);          % 取 256 个点,目的是画图的时候,维数一致
pinlv=(0:1:255)* 8000/512;             % 点和频率的对应关系
subplot(428)                           % 画第 8 个子图
plot(pinlv,signal);                    % 画幅值特性图
xlabel('频率/Hz')                      % 横坐标名称
ylabel('对数幅度/dB')                  % 纵坐标名称
title ('加汉明窗时语音谱')             % 文字标注
axis([0,4000,- 80,1])                  % 限定横纵坐标范围
```

图 4.3 为窗宽较窄的情况下清浊音波形及频谱图。语音信号采样率为 8kHz,窗长 N 取 64。由于窗很窄,选取出来的语音段的长度约 1～2 个基音周期,因而该语音短时频谱图中反映基音谐波频率的快速变化现象基本消失。但短时频谱图中仍然保留着慢变化(较宽的峰),它们是声道滤波器的共振峰。加矩形窗比加汉明时呈现出较多的细致结构,是由于矩形窗比汉明窗具有更高的频率分辨率的缘故。

综上所述,关于短时谱和移动窗可以得出以下结论。

① 长窗具有较高的频率分辨率,较低的时间分辨率。从一个基音周期到另一个基音周期,共振峰是要发生变化的,这一点即使从语音波形上也能够看出来。然而如果采用较长的窗,这种变化便被模糊了,因为长窗起到了时间上的平均作用。

② 短窗具有较低的频率分辨率,较高的时间分辨率。采用矩形窗时,能够从短时频谱中提取出共振峰从一个基音周期到另一个基音周期所发生的变化。当然,激励源的谐波的细致

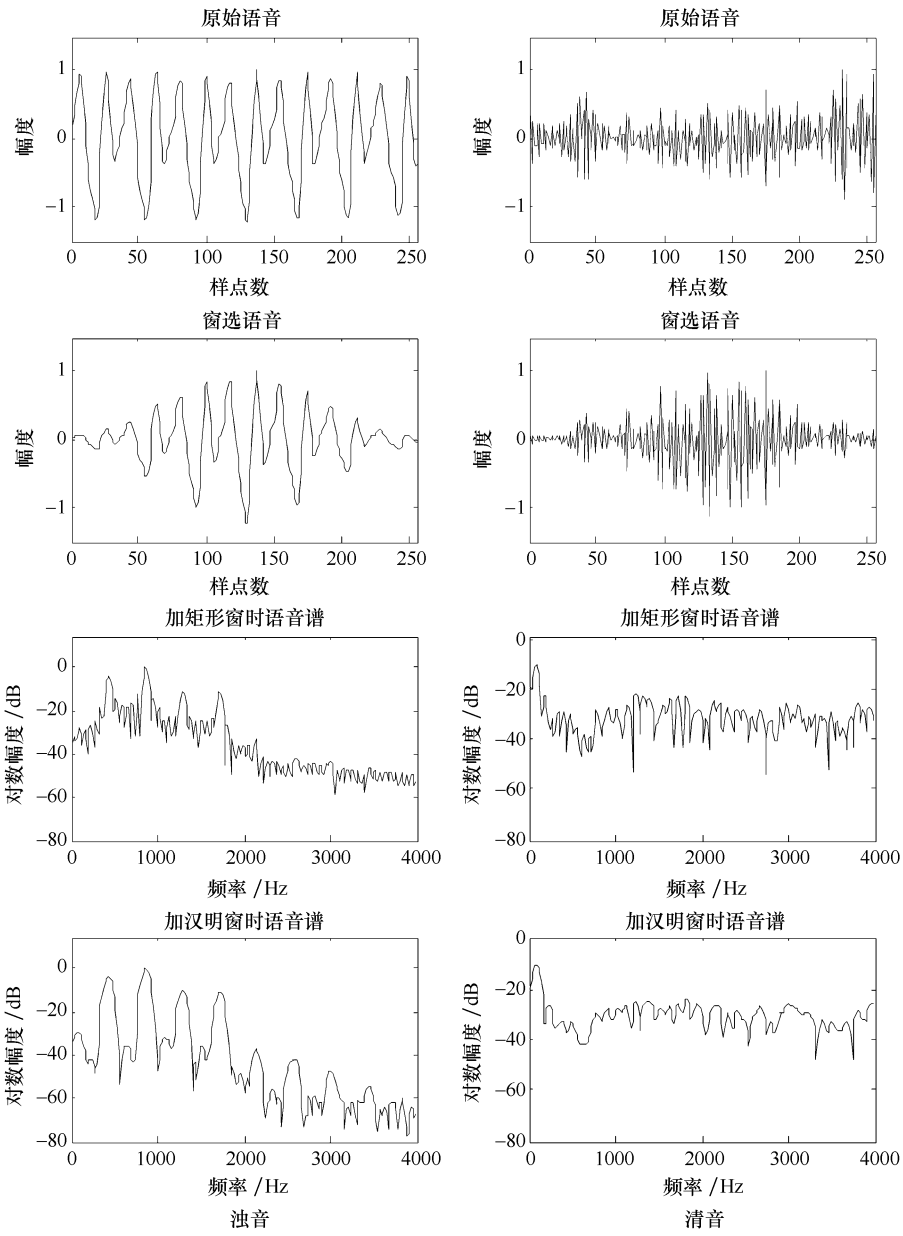


图 4.2　加不同窗函数时的清浊音波形及频谱图(窗宽 $N=256$)

结构也从短时频谱图上消失了。

③ 窗宽的选择需折中考虑。短窗具有较好的时间分辨率,能够提取出语音信号中的短时变化(这常常是分析的目的),损失了频率分辨率。但应注意到,语音信号的基音周期提取范围很大。因此,窗宽的选择应当考虑到这个因素。

④ 矩形窗和汉明窗的频谱特性都具有低通的性质,在截止频率处都比较尖锐,当其通带都比较窄时(窗越宽,其通带越窄),加窗后得到的频谱能够很好逼近短时语音信号的频谱。窗越宽,逼近效果越好。

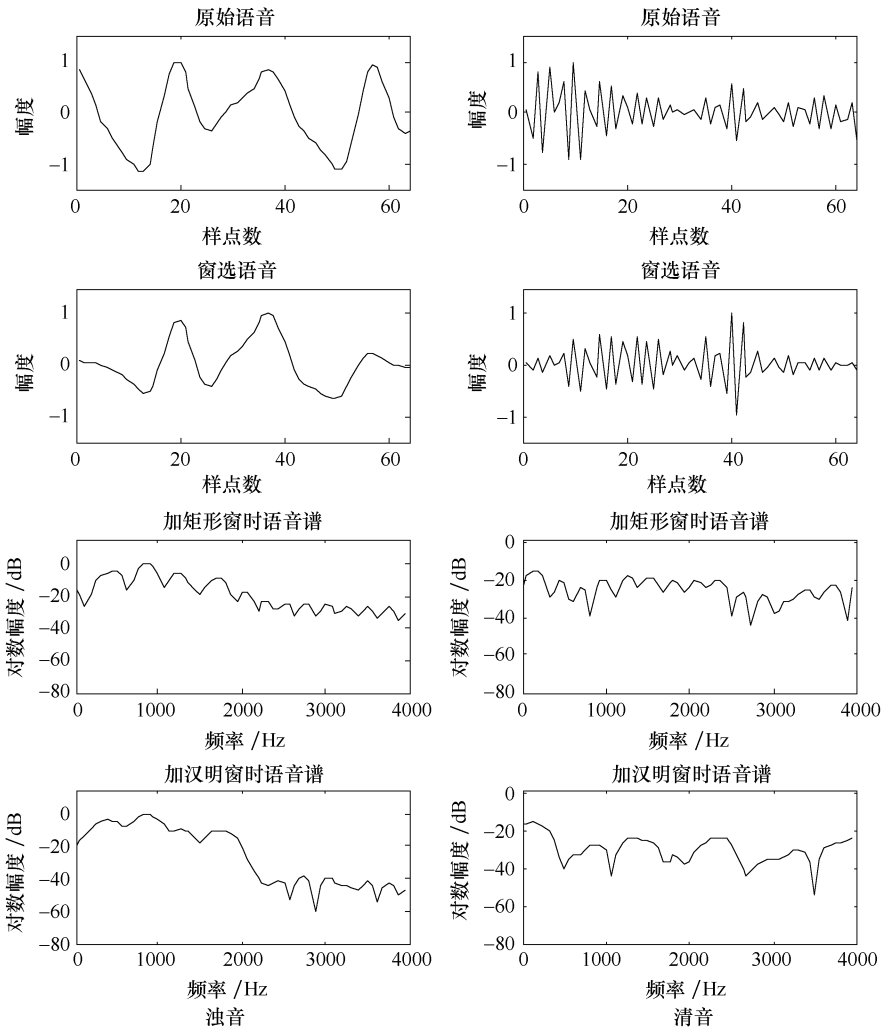


图 4.3　加不同窗函数时的清浊音波形及频谱图(窗宽 $N=64$)

4.3　滤波器的解释

1. 短时傅里叶变换的滤波器实现形式一

由式(4.1)可得

$$X_n(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{m=-\infty}^{+\infty} [x(m)\mathrm{e}^{-\mathrm{j}\omega m}]w(n-m) \tag{4.14}$$

因此,如果把 $w(n)$看做一个滤波器的单位抽样响应,则短时傅里叶变换就是设滤波器的输出为 $X_n(\mathrm{e}^{\mathrm{j}\omega})$,滤波器的输入为 $x(n)\mathrm{e}^{-\mathrm{j}\omega n}$,如图 4.4(a)所示。

因为复数可分解为实部和虚部,所以 $X_n(\mathrm{e}^{\mathrm{j}\omega})$也可以用实数来运算,即

$$X_n(\mathrm{e}^{\mathrm{j}\omega}) = |X_n(\mathrm{e}^{\mathrm{j}\omega})| \cdot \mathrm{e}^{\mathrm{j}\theta(\omega)} = a_n(\omega) - \mathrm{j}b_n(\omega) \tag{4.15}$$

其中

$$\begin{cases} a_n(\omega) = \sum_{m=-\infty}^{+\infty} x(m)\cos(\omega m)w(n-m) \\ b_n(\omega) = \sum_{m=-\infty}^{+\infty} x(m)\sin(\omega m)w(n-m) \end{cases} \tag{4.16}$$

如图 4.4(b)所示。

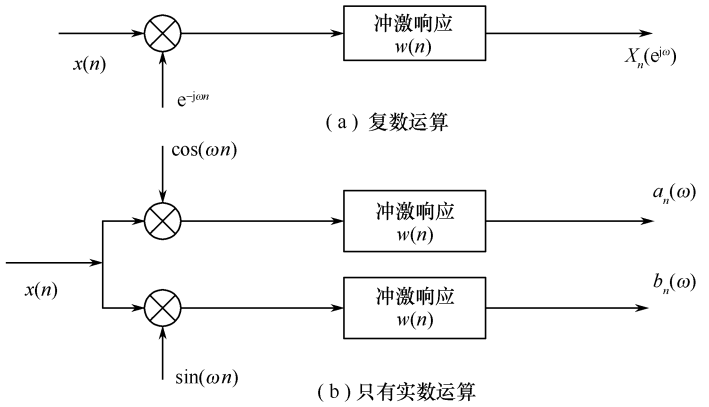


图 4.4　短时频谱分析的滤波器表示

为研究图 4.4(a)在频率 ω 上的短时傅里叶变换，假定 $x(n)$的标准傅里叶变换存在，为避免频率变量的混淆，这里将 $x(n)$的傅里叶变换写成 $X(e^{j\omega})$，将 ω 看成是某个特定的角频率值。由此可知：$x(n)$经调制后，其傅里叶变换为 $X(e^{j(\theta+\omega)})$，这说明调制使 $x(n)$的频谱在频率轴上向左移动了 ω，线性滤波器输出端的频谱等于乘积 $X(e^{j(\theta+\omega)})W(e^{j\theta})$，故为了使输出频谱准确等于 $X(e^{j\omega})$，$W(e^{j\theta})$应当是一个冲激。即要求线性滤波器近似为一个窄带低通滤波器。

2. 短时傅里叶变换的滤波器实现形式二

用滤波器来解释短时傅里叶变换还有另一种形式。令 $m'=n-m$，得

$$\begin{aligned} X_n(e^{j\omega}) &= \sum_{m'=-\infty}^{+\infty} w(m')x(n-m')e^{-j\omega(n-m')} \\ &= e^{-j\omega n}\left[\sum_{m'=-\infty}^{+\infty} x(n-m')w(m')e^{j\omega m'}\right] \end{aligned} \tag{4.17}$$

令

$$\widetilde{X}_n(e^{j\omega}) = \sum_{m'=-\infty}^{+\infty} x(n-m')w(m')e^{j\omega m'} = e^{j\omega n}X_n(e^{j\omega}) \tag{4.18}$$

则有

$$X_n(e^{j\omega}) = e^{-j\omega n} \cdot \widetilde{X}_n(e^{j\omega}) = e^{-j\omega n}[\widetilde{a}_n(\omega) - j\widetilde{b}_n(\omega)] \tag{4.19}$$

因此，可以画出短时傅里叶变换的滤波器解释的另一种形式，如图 4.5 所示，也分为图 4.5(a)复数运算和图 4.5(b)实数运算两种。

从图 4.5(a)可以看到，$X_n(e^{j\omega})$同样可被看做用复数带通滤波器的输出调制 $e^{-j\omega n}$ 的结果。此带通滤波器的冲激响应为 $w(n)e^{j\omega n}$。如果窗的傅里叶变换 $W(e^{j\theta})$是低通函数，这时图 4.5(a)中的滤波器将是一个通带中心位于 ω 频率上的窄带带通滤波器。

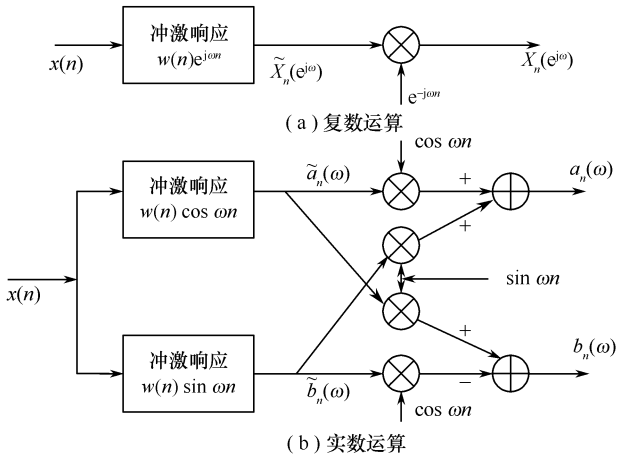


图 4.5　另一种用线性滤波对短时频谱分析的解释

4.4　短时谱的时域及频域采样率

短时傅里叶变换是一维信号 $x(n)$ 的二维表示形式，即 $X_n(e^{j\omega})$ 同时是时间 n 以及角频率 ω 的函数。如何由 $X_n(e^{j\omega})$ 来恢复 $x(n)$，首先遇到的就是时域采样率和频域采样率的问题。

1. 时域采样率

在前一节中讨论到，当 ω 为固定值时，$X_n(e^{j\omega})$ 是一个冲激响应为 $w(n)$ 的滤波器的输出，若将 $w(n)$ 的傅里叶变换记为 $W(e^{j\omega})$，对于大多数窗函数来说，$W(e^{j\omega})$ 具有低通滤波器的特性，若 $W(e^{j\omega})$ 的带宽为 BHz，$X_n(e^{j\omega})$ 则具有与窗相同的带宽。根据采样定理，$X_n(e^{j\omega})$ 的时域采样率至少为 $2B$ 才不至于发生混叠现象。低通滤波器的带宽是由 $W(e^{j\omega})$ 的第一个零点位置决定的。因为是 $w(n)$，$0\leqslant n\leqslant N-1$ 的傅里叶变换，因而 B 的取值决定于窗口序列的长度 N 和形状，因此可知，若使用汉明窗，$W(e^{j\omega})$ 的近似带宽为

$$B=\frac{2F_s}{N}(\text{Hz}) \tag{4.20}$$

其中，F_s 是信号 $x(n)$ 的采样率，因此，$X_n(e^{j\omega})$ 在时间域内要求的采样率为 $2B=\dfrac{4F_s}{N}$。

2. 频率采样率

当 n 为固定值时，$X_n(e^{j\omega})$ 是序列 $x(n)w(n-m)$ 的傅里叶变换。为得到 $x(n)$，必须对 $X_n(e^{j\omega})$ 进行 ω 域的采样，再由离散傅里叶反变换求出 $x(n)$，这时我们利用了 $X_n(e^{j\omega})$ 的傅里叶变换的解释。由于 $X_n(e^{j\omega})$ 是以 2π 为周期的 ω 的连续函数，只需在 2π 长度间隔内采样。

设窗为时间受限的，因为把 $X_n(e^{j\omega})$ 看做一个傅里叶变换，则它的反变换也在时间上受限。根据采样定理要求，我们在频域内对 $X_n(e^{j\omega})$ 用至少两倍于它的“时间宽度”的速率来采样。$X_n(e^{j\omega})$ 的傅里叶反变换是序列 $x(n)w(n-m)$，其中窗 $w(n)$ 的宽度为 N，所以信号 $x(n)w(n-m)$ 的宽度等于 N 个采样。为了从 $X_n(e^{j\omega})$ 中恢复 $x(n)$，$X_n(e^{j\omega})$ 必须用下述一组频率值来采样

$$\omega_k=\frac{2\pi k}{N},\quad k=0,1,\cdots,N-1 \tag{4.21}$$

3. 总采样率

基于上述讨论，我们能确定每秒内使原始信号 $x(n)$ 得到非混叠表示所必须的 $X_n(e^{j\omega})$ 的采样总数。$X_n(e^{j\omega})$ 在时间域内的最小采样率为 $2B$，其中 B 是窗的频带宽度，而频率域内的最小采样为 N，即为窗宽。因此，$X_n(e^{j\omega})$ 的总采样率(SR)等于

$$\mathrm{SR}=2B\cdot N(\text{采样/秒}) \tag{4.22}$$

在大多数实际窗中，B 可以表示为 F_s/N 的倍数，即

$$B=C\frac{F_s}{N}\quad(\mathrm{Hz}) \tag{4.23}$$

其中，C 是比例常数，式(4.23)代入式(4.22)中，得

$$\mathrm{SR}=2CF_s(\text{采样/秒}) \tag{4.24}$$

式中，SR/F_S 即为与一般采样频率相比而得到的“过速率采样比”。若 $w(n)$ 使用汉明窗，则 $2C=4$；若采用矩形窗，则 $2C=2$，因此，$x(n)$ 的短时谱所要求的采样率比起一般波形表示来说，要增加到 2～4 倍。但有时在时域或频域用低于理论上最小值的采样率，而 $x(n)$ 仍能从混叠的短时变换中准确地恢复，这称为欠速率采样。欠速率采样在短时谱估计、基音及共振峰分析、数字语谱图及声码器中得到应用。

4.5 短时综合的滤波器组相加法

前面讨论了语音的短时傅里叶分析方法以及短时傅里叶变换在时域和频域的采样。本节讨论如何从短时傅里叶变换的采样恢复原始语音信号的问题，通常称为语音的短时合成。常用的短时合成技术有两种：滤波器组相加法和叠接相加法。本节仅讨论前者，下一节将讨论后者。

4.5.1 短时综合的滤波器组相加法原理

滤波器组相加法是利用滤波器组表示语音的短时谱的方法。由式(4.1)知，可将 $X_n(e^{j\omega})$ 表示为

$$X_n(e^{j\omega_i})=\sum_{m=-\infty}^{+\infty}w_i(n-m)x(m)e^{-j\omega_i m} \tag{4.25}$$

或

$$X_n(e^{j\omega_i})=e^{-j\omega_i n}\sum_{m=-\infty}^{+\infty}x(n-m)w_i(m)e^{j\omega_i m} \tag{4.26}$$

式中，$w_i(m)$ 是在频率 ω_i 上使用的窗，若定义

$$h_i(n)=w_i(n)e^{j\omega_i n} \tag{4.27}$$

则式(4.26)可以表示为

$$X_n(e^{j\omega_i})=e^{-j\omega_i n}\sum_{m=-\infty}^{+\infty}x(n-m)h_i(m) \tag{4.28}$$

由于窗 $w_i(n)$ 具有低通滤波特性，式(4.28)可以理解为先经过一个冲激响应为 $h_i(n)$ 的带通滤波器，再用复指数 $e^{-j\omega_i n}$ 调制，如图 4.6 所示。

若定义

$$y_i(n)=X_n(\mathrm{e}^{\mathrm{j}\omega_i})\mathrm{e}^{\mathrm{j}\omega_i n} \tag{4.29}$$

则由式(4.28)可得

$$y_i(n)=\sum_{m=-\infty}^{+\infty}x(n-m)h_i(m) \tag{4.30}$$

由式(4.30)可见，$y_i(n)$是一个冲激响应为$h_i(n)$的带通滤波器的输出。$h_i(n)$由式(4.27)决定。图 4.6(a)表示式(4.28)与式(4.27)的运算过程，图 4.6(b)表示式(4.25)和式(4.27)的运算过程，图 4.6(c)表示了两种情况下的等效带通滤波器。

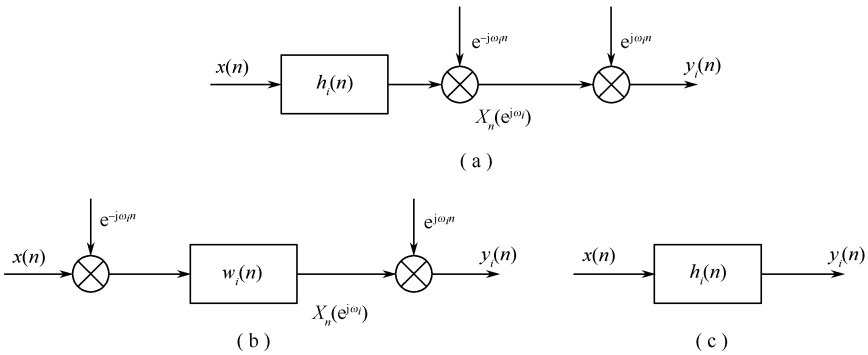


图 4.6　用线性滤波实现单个通道综合的方法

利用上面讨论的结果，可以获得恢复输入信号的实际方法。考虑有 N 个满足式(4.27)的带通滤波器，其中对于 $i=0,1,\cdots,N-1$，共有 N 个频率 $\omega_i=\dfrac{2\pi}{N}i$，假定 $w_i(n)$是一个截止频率为 ω_{pi} 的理想低通滤波器的冲激响应时，图 4.7(a)表示此滤波器的频率响应 $W_i(\mathrm{e}^{\mathrm{j}\omega})$，对应的复数带通滤波器的冲激响应如式(4.27)所示，其频率响应为

$$H_i(\mathrm{e}^{\mathrm{j}\omega})=W_i(\mathrm{e}^{\mathrm{j}(\omega-\omega_i)}) \tag{4.31}$$

式(4.31)用图 4.7(b)表示，中心频率为 ω_i，带宽为 $2\omega_{pi}$，假定所有通道都使用了相同的窗函数，即

$$w_i(n)=w(n),\quad i=0,1,\cdots,N-1 \tag{4.32}$$

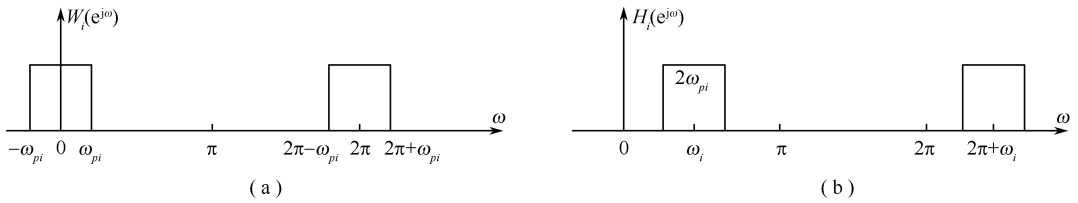


图 4.7　(a)理想低通滤波器的频率响应；
(b)理想带通滤波器的频率响应

考虑整个带通滤波器组时，其中每个带通滤波器具有相同的输入，其输出相加在一起，如图 4.8 所示，输出为 $y(n)$，输入为 $x(n)$，整个系统的复合频率响应为

$$\widetilde{H}(\mathrm{e}^{\mathrm{j}\omega})=\sum_{i=0}^{N-1}H_i(\mathrm{e}^{\mathrm{j}\omega})=\sum_{i=0}^{N-1}W(\mathrm{e}^{\mathrm{j}(\omega-\omega_i)}) \tag{4.33}$$

如果 $W(\mathrm{e}^{\mathrm{j}\omega})$在频率域上正确采样($N\geqslant L$，$L$ 为窗宽)可以证明对于所有 ω 都满足

$$\frac{1}{N}\sum_{i=0}^{N-1}W(\mathrm{e}^{\mathrm{j}(\omega-\omega_i)})=w(0) \tag{4.34}$$

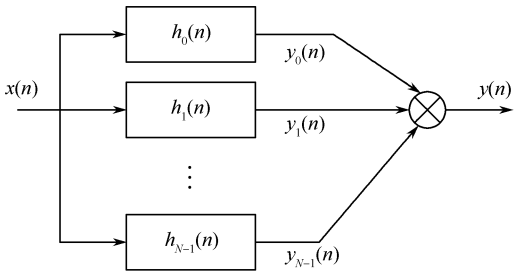


图 4.8　用带通滤波器将 $y(n)$与 $x(n)$联系起来

上式证明如下：

$W(\mathrm{e}^{\mathrm{j}\omega})$的傅里叶反变换是窗函数 $w(n)$，如果 $W(\mathrm{e}^{\mathrm{j}\omega})$在频率上以 N 个均匀间隔采样，则 $W(\mathrm{e}^{\mathrm{j}\omega_i})$采样形式的离散傅里叶反变换为

$$\frac{1}{N}\sum_{i=0}^{N-1}W(\mathrm{e}^{\mathrm{j}\omega_i})\mathrm{e}^{\mathrm{j}\omega_i n}=\sum_{r=-\infty}^{+\infty}w(n+rN) \tag{4.35}$$

如果 $w(n)$的宽度等于 L 个采样，则

$$w(n)=0,\quad n<0,\ n\geqslant L \tag{4.36}$$

这时只要 $W(\mathrm{e}^{\mathrm{j}\omega})$在频率域上正确采样($N\geqslant L$)，就不会引起混叠。在式(4.35)中取 $n=0$，得

$$\frac{1}{N}\sum_{i=0}^{N-1}W(\mathrm{e}^{\mathrm{j}\omega_i})=w(0) \tag{4.37}$$

考虑到 $W(\mathrm{e}^{\mathrm{j}(\omega-\omega_i)})$是 $W(\mathrm{e}^{\mathrm{j}\omega})$在 $\omega-\omega_i$ 上而不是在 ω_i 上的均匀采样形式后，我们能得出式(4.34)，因为按采样定理，任何一组 N 个均匀分布的采样都是适用的。

由式(4.27)和式(4.34)可以推出复合系统的冲激响应为

$$\tilde{h}(n)=\sum_{i=0}^{N-1}h_i(n)=\sum_{i=0}^{N-1}w(n)\mathrm{e}^{\mathrm{j}\omega_i n}=Nw(0)\delta(n) \tag{4.38}$$

这时的复合输出为

$$y(n)=Nw(0)x(n) \tag{4.39}$$

于是，用滤波器组相加法恢复的信号可以表示为

$$y(n)=\sum_{i=0}^{N-1}y_i(n)=\sum_{i=0}^{N-1}X_n(\mathrm{e}^{\mathrm{j}\omega_i})\mathrm{e}^{\mathrm{j}\omega_i n} \tag{4.40}$$

式(4.40)中所包含的分析与综合运算过程如图 4.9 所示，其中的滤波器均为带通滤波器。

上面的讨论说明，当 $w(n)$具有有限宽度 L 时，$x(n)$完全能从时间及频率域采样后的时变傅里叶变换准确地恢复。也可以证明，如果 $W(\mathrm{e}^{\mathrm{j}\omega})$在频域内是频带受限的，则 $x(n)$也能准确从 $X_n(\mathrm{e}^{\mathrm{j}\omega_i})$中恢复，这里证明从略。

4.5.2　短时综合的滤波器组相加法的 MATLAB 程序实现

程序 4.2 对应于图 4.6(b)，先调制后滤波，实现流程图如图 4.10 所示，程序运行结果如图 4.11 所示。程序 4.2 对应于图 4.6(a)，先滤波后调制，实现流程图如图 4.12 所示，程序运行结果如图 4.13 所示。

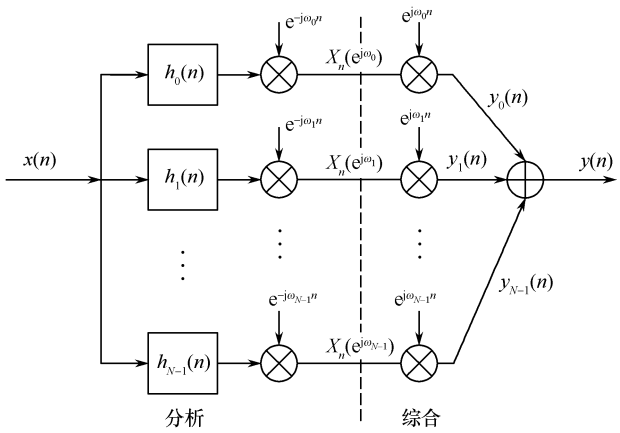


图 4.9　短时频谱中的分析与综合运算过程

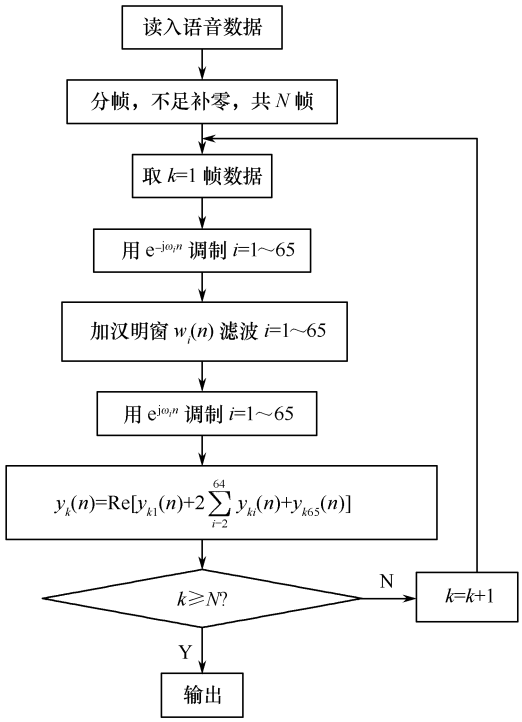


图 4.10　程序 4.2 的实现流程图

【程序 4.2】filterbank1.m

```
clear; clf;
WL= 256;                                % 窗长
N=128;                                  % 滤波器通道个数
M=1024;                                 % 语音帧的大小,必须是窗长的倍数
[IN, FS] = wavread('speech.wav');       % 读入一段语音,FS 为采样率
L= length(IN);                          % 输入语音的长度
window = hann(WL);                      % Hanning 窗,窗长为 WL
% * * * * * * * 将语音分帧,每帧大小为 M,若语音长度不是 M 的整数倍
```

```
% * * * * * * * 则需补零至能整除为止并将语音幅度归一化
Mod=M- mod(L,M);
Q=(L+ Mod)/M;                          % 补零后的语音帧数
IN=[IN;zeros(Mod,1)]/max(abs(IN));
% ************************所需变量的初始化*****************************
OUT=zeros(length(IN),1);
X=zeros(M,(N/2+ 1));
Z=zeros(WL- 1,(N/2+ 1));
t= (0:M- 1)';
WN1= zeros(M,(N/2+ 1));
WN2= zeros(M,(N/2+ 1));
% ************************************************************************
for k=1:(N/2+ 1)
w=2* pi* i* (k- 1)/N;                  % 各个通道的一组角频率
WN1(:,k)=exp(- w* t);
WN2(:,k)=exp(w* t);
end
for p=1:Q;
R=IN((p- 1)* M+ 1:p* M);               % 每次取一帧语音,直至将语音取完
for k=1:(N/2+ 1)
        x=R.* WN1(:,k);                % 对取进来的语音进行调制
[X(:,k), Z(:,k)] = filter(window, 1, x, Z(:,k));% 将调制后的语音进行加窗滤波
end
X1= X.* WN2;                           % 将滤波后的信号进行反调制
% 由于对取进来的语音进行调制时会发现,第 2 个通道与第 128 个通道,第 3 通道与
% 第 127 通道,…第 64 通道与第 66 通道共轭,因此计算时只需计算前 65 个通道的
% 滤波和反调制结果,最后的输出等于第 2 至 64 通道输出结果的实部的 2 倍之和加上
% 第 1 通道和第 65 通道的实部

    A=zeros(M,1);
    for j=2:(N/2)
        A=A+ X1(:,j);
    end
    Y((p- 1)* M+ 1:p* M)=2* real(A)+ real(X1(:,1)+ X1(:,65));
       Y1((p- 1)* M+ 1:p* M)=real(X1(:,1));
       Y2((p- 1)* M+ 1:p* M)=real(X1(:,2));
       Y65((p- 1)* M+ 1:p* M)=real(X1(:,65));
end
%
OUT =Y(1:L) / max(abs(Y));  % 输出语音幅度归一化
wavwrite(OUT, FS, 'wn.wav');% 将 OUT 写入 wav 文件 wn
wavplay(OUT,FS);                       % 播放 wn.wav 文件
% 绘出输入与输出语音的时域波形图并显示在一幅图中
figure(1);
```

```
subplot(511);
plot(IN);
title('输入语音');
xlabel('样点数');
ylabel('幅度');
subplot(512);
plot(Y1);
title('第 1 通道输出语音');
xlabel('样点数');
ylabel('幅度');
subplot(513);
plot(Y2);
title('第 2 通道输出语音');
xlabel('样点数');
ylabel('幅度');
subplot(514);
plot(Y65);
title('第 65 通道输出语音');
xlabel('样点数');
ylabel('幅度');
subplot(515);
plot(OUT);
title('输出语音');
xlabel('样点数');
ylabel('幅度');
```

程序运行结果如图 4.11 所示。

【程序 4.3】 filterbank2.m

```
clear; clf;
WL= 256;                                % 窗长
N=128;                                  % 滤波器通道个数
M=1024;                                 % 语音帧的大小,必须是窗长的倍数
[IN, FS] = wavread('speech.wav');       % 读入一段语音,FS 为采样率
L= length(IN);                          % 输入语音的长度
window = hann(WL);                      % Hanning 窗,窗长为 WL
% * * * * * * * 将语音分帧,每帧大小为 M,若语音长度不是 M 的整数倍
% * * * * * * * 则需补零至能整除为止并将语音幅度归一化
Mod=M- mod(L,M);
Q=(L+ Mod)/M;                           % 补零后的语音帧数
IN=[IN;zeros(Mod,1)]/max(abs(IN));
% ************************所需变量的初始化*****************************
OUT=zeros(length(IN),1);
X=zeros(M, (N/2+ 1));
Z=zeros(WL- 1, (N/2+ 1));
t= (- WL/2:WL/2- 1)';
```

```
WN＝zeros(WL, (N/2+ 1));
% ************************************************************************
for k＝1:(N/2+ 1)
w＝2* pi* i* (k- 1)/N;                  % 各个通道的一组角频率
    WN(:,k)＝exp(w* t);
end
for p＝1:Q;
x＝IN((p- 1)* M+ 1:p* M);               % 每次取一帧语音,直至将语音取完
% 将取进来的语音加窗调制滤波
    for k＝1:(N/2+ 1)
       [X(:,k), Z(:,k)] ＝ filter(window.* WN(:,k), 1, x, Z(:,k));
    end

% 由于对取进来的语音进行加窗调制滤波时会发现,第 2 个通道与第 128 个通道
% 第 3 通道与第 127 通道…第 64 通道与第 66 通道共轭,因此在计算时只需计算前 65 个通道
% 的滤波和反调制结果
% 最后的输出等于第 2 至 64 通道输出结果的实部的 2 倍之和加上第 1 通道和 65 通道的实部
    A＝zeros(M,1);
    for j＝2:(N/2)
        A＝A+ X(:,j);
    end
Y((p- 1)* M+ 1:p* M)＝2* real(A)+ real(X(:,1)+ X(:,65));
    Y1((p- 1)* M+ 1:p* M)＝real(X(:,1));
    Y2((p- 1)* M+ 1:p* M)＝real(X(:,2));
    Y65((p- 1)* M+ 1:p* M)＝real(X(:,65));
end
%
OUT ＝Y(1:L) / max(abs(Y));             % 输出语音幅度归一化
wavwrite(OUT, FS, 'wn.wav');            % 将 OUT 写入 wav 文件 wn
wavplay(OUT,FS);                        % 播放 wn.wav 文件
% 绘出输入与输出语音的时域波形图并显示在一幅图中
figure(1);
subplot(511);
plot(IN);
title('输入语音');
xlabel('样点数');
ylabel('幅度');
subplot(512);
plot(Y1);
title('第 1 通道输出语音');
xlabel('样点数');
ylabel('幅度');
subplot(513);
plot(Y2);
```

```
title('第 2 通道输出语音');
xlabel('样点数');
ylabel('幅度');
subplot(514);
plot(Y65);
title('第 65 通道输出语音');
xlabel('样点数');
ylabel('幅度');
subplot(515);
plot(OUT);
title('输出语音');
xlabel('样点数');
ylabel('幅度');
```

程序运行结果如图 4.13 所示。

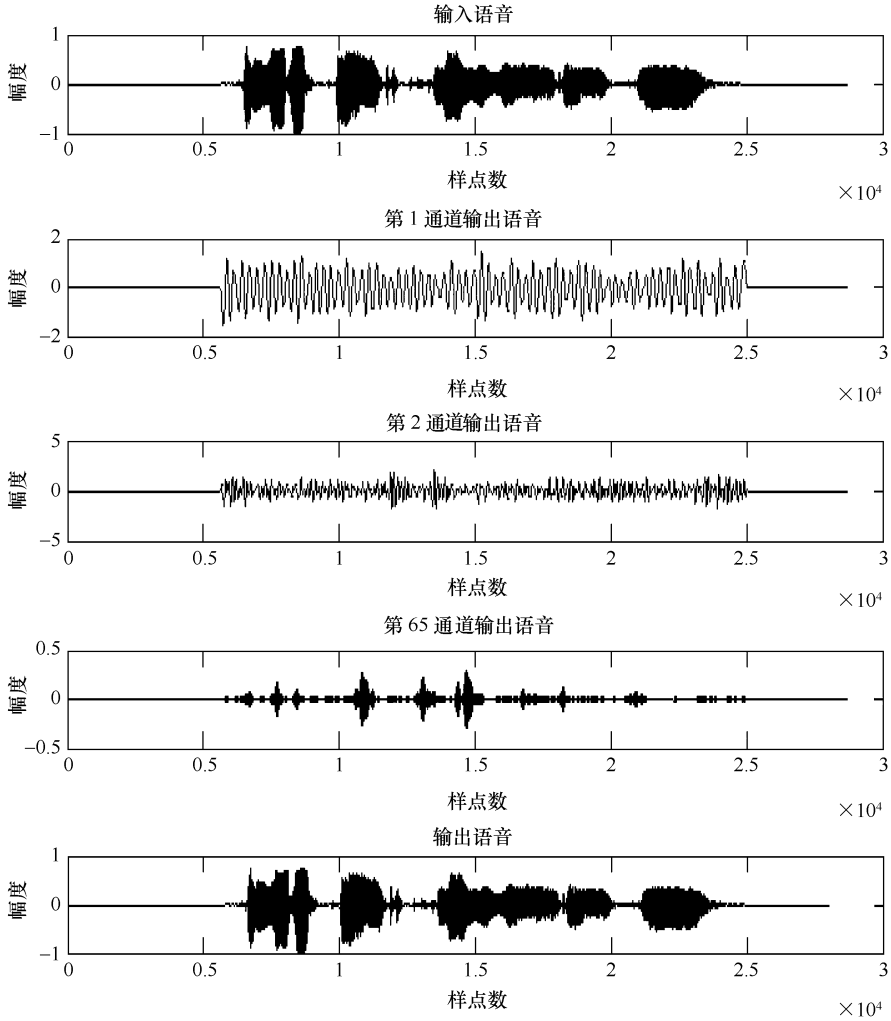


图 4.11　程序 4.2 的运行结果

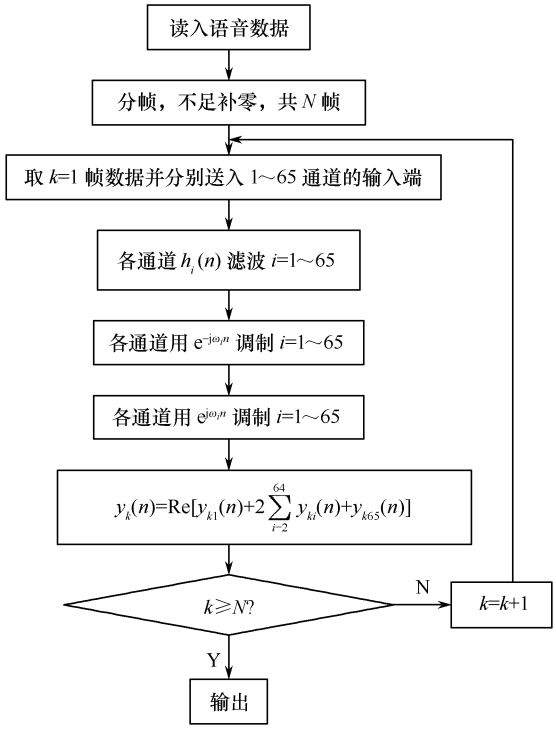


图 4.12　程序 4.3 的实现流程图

4.5.3　短时综合的叠接相加法原理及 MATLAB 程序实现

假设在时域上利用周期为 R 的取样对 $X_n(e^{j\omega_i})$进行取样，则得

$$Y_r(e^{j\omega_i})=X_n(e^{j\omega_i})\big|_{n=rR}=X_{rR}(e^{j\omega_i}) \tag{4.41}$$

式中，r 为一整数，$0\leqslant i\leqslant N-1$，求上式的反变换，可得

$$y_r(n)=\frac{1}{N}\sum_{i=0}^{N-1}Y_r(e^{j\omega_i})e^{j\omega_i n} \tag{4.42}$$

又

$$y_r(k)=x(k)w(rR-k),\quad (-\infty<k<\infty) \tag{4.43}$$

因而

$$y(n)=\sum_{r=-\infty}^{+\infty}y_r(n)=x(n)\sum_{r=-\infty}^{+\infty}w(rR-n) \tag{4.44}$$

将式(4.42)代入式(4.44)中，可得

$$y(n)=\sum_{r=-\infty}^{+\infty}\left[\frac{1}{N}\sum_{i=0}^{N-1}Y_r(e^{j\omega_i})e^{j\omega_i n}\right] \tag{4.45}$$

如果 $w(n)$的傅里叶变换频带受限且 $X_n(e^{j\omega_i})$在时域上被正确取样，即 R 选得足够小，这时不论 n 为何值均可写出

$$\sum_{r=-\infty}^{+\infty}w(rR-n)=\sum_{r=-\infty}^{+\infty}w(rR-n)e^{j(rR-0)0}\approx W(e^{j0})/R \tag{4.46}$$

因而，式(4.44)可写成

$$y(n)=x(n)W(e^{j0})/R \tag{4.47}$$

上式说明，$y(n)$与 $x(n)$只差一个常系数，因而利用式(4.45)就能准确恢复 $x(n)$，图 4.14 为短

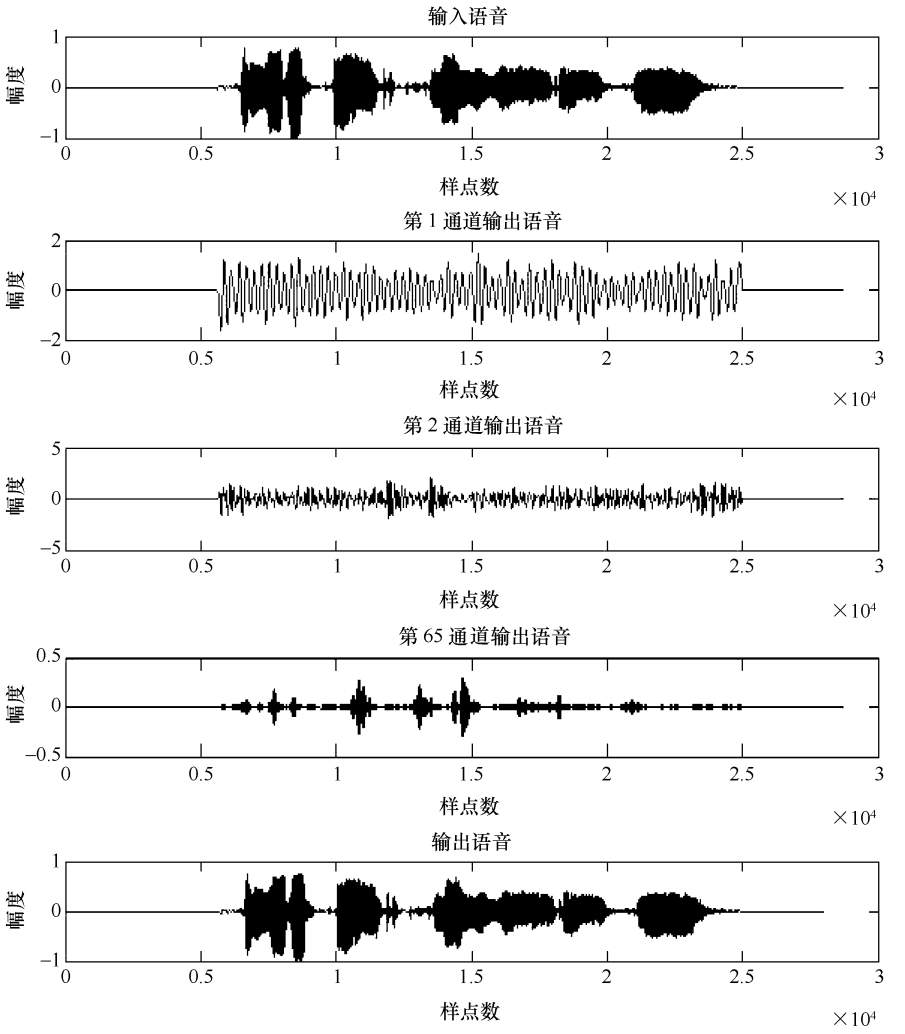


图 4.13　程序 4.3 的运行结果

时综合叠接相加法的流程图。图 4.15 表示利用一个 L 点汉明窗计算 $y(n)$的过程。

在图 4.14 及图 4.15 的例子中，假定 $n<0$ 时 $x(n)=0$，对汉明窗需要 4∶1 的时间重叠，即 $R=\dfrac{L}{4}$在图 4.15 中，第一分析段从 $n=\dfrac{L}{4}$为标志，利用窗(窗宽为 L)来得到信号。

$$y_r(k)=x(k)w(rR-k) \tag{4.48}$$

此时信号在 $rR-L+1\leqslant k\leqslant rR$ 范围内不为零，填充零值后，得到 N 点序列，求 N 点 FFT 即可求得 $Y_r(\mathrm{e}^{\mathrm{j}\omega_i})$。

图 4.15 表示了按照式(4.44)的运算过程，当 $0\leqslant n\leqslant R-1$ 时，$y(n)$可写成

$$\begin{aligned} y(n)=&x(n)w(R-n)+x(n)w(2R-n)+\\ &x(n)w(3R-n)+x(n)w(4R-n) \end{aligned} \tag{4.49}$$

当 $R\leqslant n\leqslant 2R-1$ 时，则 $y(n)$可写成

$$\begin{aligned} y(n)=&x(n)w(2R-n)+x(n)w(3R-n)+\\ &x(n)w(4R-n)+x(n)w(5R-n) \end{aligned} \tag{4.50}$$

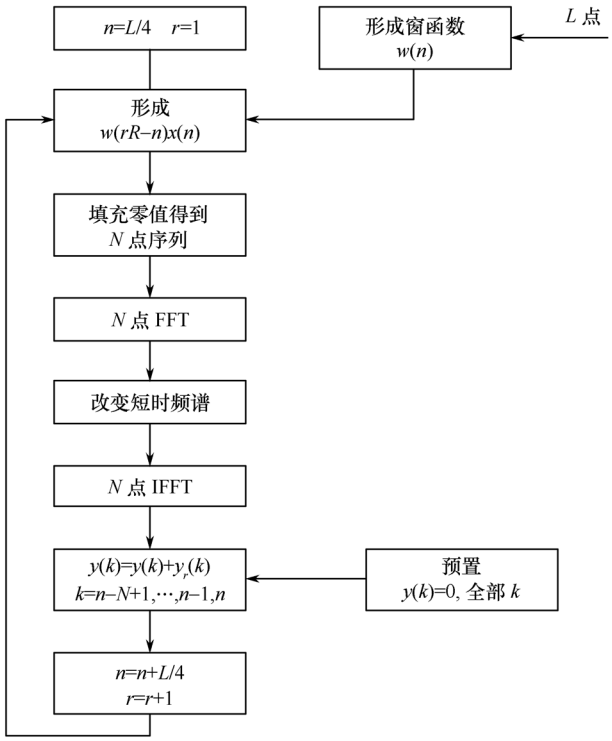


图 4.14　短时综合叠接相加法流程图

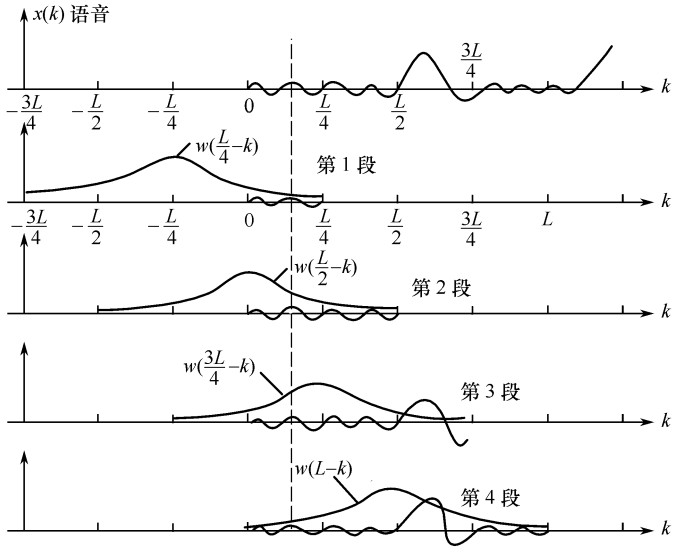


图 4.15　利用一个 L 点汉明窗时 $y(n)$的计算过程

滤波器组相加法与频率取样有关,它所要求的频率取样数应使窗变换满足

$$\frac{1}{N}\sum_{i=0}^{N-1} W(\mathrm{e}^{\mathrm{j}(\omega-\omega_i)}) = w(0) \tag{4.51}$$

而重叠相加法要求时间抽样率应选得使窗满足

$$\sum_{r=-\infty}^{+\infty} w(rR-n) = W(e^{j0})/R \tag{4.52}$$

式(4.51)与式(4.42)构成对偶关系。

下面给出了短时综合的叠接相加法的 MATLAB 实现程序。

【程序 4.4】 ShortTimeAdd.m

```
clear all;
s=wavread('speech.wav');          % 读入一段语音
s=s';                             % 将 s 转置
N=length(s);                      % 读入语音的长度
L=280;                            % 窗长
R=L/4;                            % 帧长
w=hamming(L);                     % 汉明窗
w=w';                             % 将 w 转置
k=((N- mod(N,R))/R);              % 如 N 不是 R 的倍数,将最后剩余的去掉不作处理
                                  % 取一帧语音,直至取完
for i=0:k- 1
for n=(1+ i* R):((i+ 1)* R)
    y(n)=s(n)* (w((i+ 1)* R- n+ 1)+ w((i+ 2)* R- n+ 1)+ w((i+ 3)* R- n+ 1)+ w((i
+ 4)* R- n+ 1));
end
b=[y((1+ i* R):((i+ 1)* R)),zeros(1,3* R)];  % 给 y 补 3R 个零,使达到 L 点
c=fft(b,L);                       %  对 b 进行 L 点傅里叶变换
d=ifft(c,L);                      % 对 c 进行 L 点傅里叶逆变换
e((1+ i* R):((i+ 1)* R))=d(1:R);  % 存储数据
end
e=e/max(abs(e));
wavwrite(e,'wnt.wav');            % 将 e 写入 wav 文件 wnt
wavplay(e,8000);                  % 播放 wnt 文件
                                  % 绘图
figure(1);
subplot(2,1,1);
plot(s);
title('输入语音');
xlabel('样点数');
ylabel('幅度');
subplot(2,1,2);
plot(e);
title('输出语音');
xlabel('样点数');
ylabel('幅度');
```

程序运行结果如图 4.16 所示。

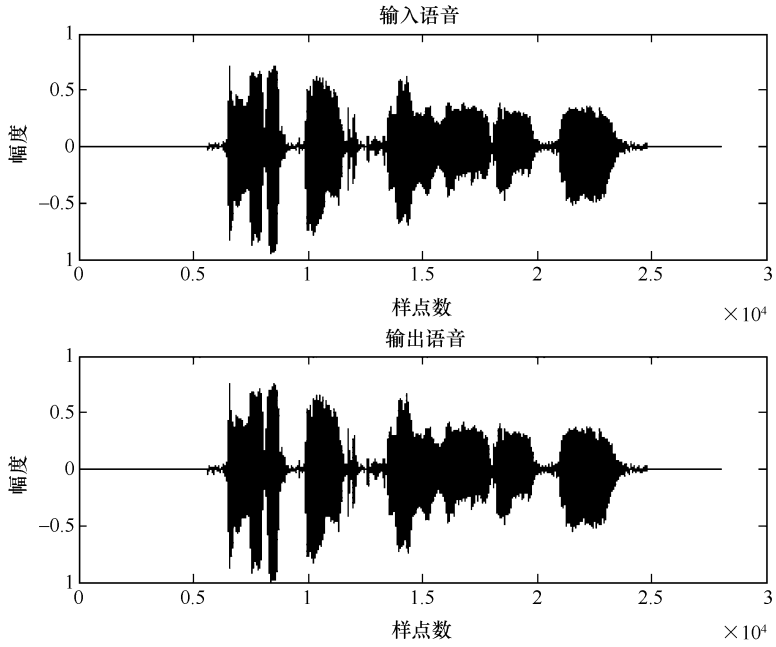


图 4.16　程序 4.4 的运行结果

第 5 章　语音信号的同态处理

5.1　概　　述

同态处理方法是一种设法将非线性问题转化为线性问题来进行处理的方法，它能将两个信号通过乘法合成的信号，或通过卷积合成的信号分开。对于语音信号，我们的目的是要从声道冲激响应分量与激励分量的卷积中分开各原始分量。由卷积结果求得参与卷积的各信号分量是涉及数字信号处理理论的一项任务，称为“解卷积”或简称“解卷”。

对语音信号进行同态分析后，将得到语音信号的倒谱参数，因此同态分析也称为倒谱分析或同态处理。

5.2　叠加原理和广义叠加原理

对于一个线性系统来说，其输入输出的关系服从叠加原理。设输入信号 $x(n)$ 是由两个信号分量 $x_1(n)$、$x_2(n)$ 的和构成，系统输出为 $y(n)$，则有

$$\begin{aligned} L[x(n)]&=L[x_1(n)+x_2(n)]=L[x_1(n)]+L[x_2(n)]\\ &=y_1(n)+y_2(n)=y(n) \end{aligned} \tag{5.1}$$

$$L[ax(n)]=aL[x(n)]=ay(n) \tag{5.2}$$

其中，L 表示线性算子。叠加原理可以简述如下：如果输入信号是若干基元信号的线性组合，则系统输出是各个对应系统的线性组合。

通过模仿普通线性系统的叠加原理，我们能定义一类系统，它服从广义叠加原理，其中加法可由卷积代替。即有

$$\begin{aligned} H[x(n)]&=H[x_1(n)*x_2(n)]=H[x_1(n)]*H[x_2(n)]\\ &=y_1(n)*y_2(n)=y(n) \end{aligned} \tag{5.3}$$

同理也可给出一个类似于式(5.2)的广义标量相乘的公式，但这个概念对于我们后面讨论的应用并不必要。因此，如果一个系统具有式(5.3)所表示的性质，则称为“卷积同态系统”。

5.3　卷积同态系统

如图 5.1 所示为卷积同态系统示意图。卷积同态系统的典范表示如图 5.2 所示，它由三部分组成：特征系统 $D_*[\ ]$、线性系统 L 及逆特征系统 $D_*^{-1}[\ ]$。

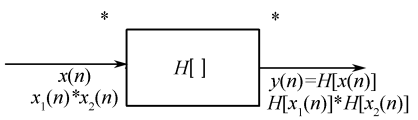


图 5.1　卷积同态系统模型

第一部分为特征系统 $D_*[\ ]$，其输入是若干信号的卷积组合，而输出为若干信号的加法组合。特征系统 $D_*[\ ]$ 有下述性质：

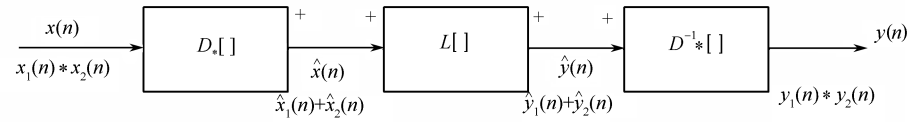


图 5.2　卷积同态系统的典范表示

$$D_*[x(n)]=D_*[x_1(n)*x_2(n)]=D_*[x_1(n)]+D_*[x_2(n)]$$
$$=\hat{x}_1(n)+\hat{x}_2(n)=\hat{x}(n) \tag{5.4}$$

第二部分是一个普通的线性系统，它服从一般的叠加原理，如下式表示：

$$L[\hat{x}(n)]=L[\hat{x}_1(n)+\hat{x}_2(n)]=L[\hat{x}_1(n)]+L[\hat{x}_2(n)]$$
$$=y_1(n)*y_2(n)=y(n) \tag{5.5}$$

第三部分是特征系统 $D_*[\ ]$ 的逆系统，它将信号的加法组合变换回卷积组合。逆特征系统 D_*^{-1} 有下述性质：

$$D_*^{-1}[\hat{y}(n)]=D_*^{-1}[\hat{y}_1(n)+\hat{y}_2(n)]=D_*^{-1}[\hat{y}_1(n)]*D_*^{-1}[\hat{y}_2(n)]$$
$$=y_1(n)*y_2(n)=y(n) \tag{5.6}$$

按照卷积定理，时域上是两个信号的卷积，则其 z 变换是两个信号 z 变换的乘积，即：

$$x(n)=x_1(n)*x_2(n) \tag{5.7}$$

其 z 变换为

$$X(z)=X_1(z)\cdot X_2(z) \tag{5.8}$$

利用 z 变换表示，卷积组合可变为乘法组合，再利用对数特性，可将乘法组合变为加法组合，再进行逆 z 变换，输出信号仍为加法组合，这就构成了卷积同态系统的特征系统 $D_*[\ ]$，如图 5.3 所示。其中

$$\hat{X}(z)=\ln X(z)=\ln[X_1(z)\cdot X_2(z)]$$
$$=\ln[X_1(z)]+\ln[X_2(z)]=\hat{X}_1(z)+\hat{X}_2(z) \tag{5.9}$$

$$z^{-1}[\hat{X}(z)]=z^{-1}[\hat{X}_1(z)]+z^{-1}[\hat{X}_2(z)]=\hat{x}_1(n)+\hat{x}_2(n)=\hat{x}(n) \tag{5.10}$$

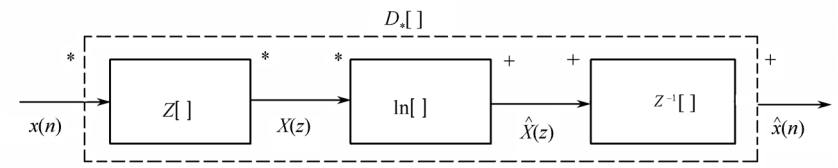


图 5.3　卷积同态系统的特征系统

卷积同态系统的逆特征系统 D_*^{-1} 如图 5.4 所示。其中

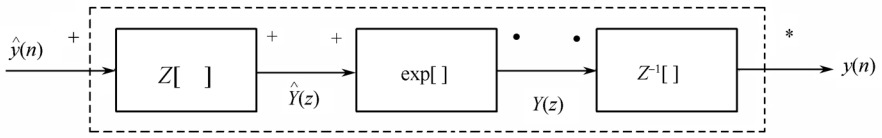


图 5.4　卷积同态系统的逆特征系统

$$z[\hat{y}(n)]=z[\hat{y}_1(n)+\hat{y}_2(n)]=z[\hat{y}_1(n)]+z[\hat{y}_2(n)]$$
$$=\hat{Y}_1(z)+\hat{Y}_2(z)=\hat{Y}(z) \tag{5.11}$$

$$\exp[\hat{Y}(z)]=\exp[\hat{Y}_1(z)+\hat{Y}_2(z)]=\exp[\hat{Y}_1(z)]\cdot\exp[\hat{Y}_2(z)]$$
$$=Y_1(z)\cdot Y_2(z)=Y(z) \tag{5.12}$$
$$z^{-1}[Y(z)]=z^{-1}[Y_1(z)\cdot Y_2(z)]=y_1(n)*y_2(n)=y(n) \tag{5.13}$$

5.4 复倒谱和倒谱

5.4.1 定义

设信号 $x(n)$ 的 z 变换为 $X(z)=z[x(n)]$，其对数为

$$\hat{X}(z)=\ln X(z)=\ln[z[x(n)]] \tag{5.14}$$

那么 $\hat{X}(z)$ 的逆 z 变换可写成

$$\hat{x}(n)=z^{-1}[\hat{X}(z)]=z^{-1}[\ln X(z)]=z^{-1}[\ln z[x(n)]] \tag{5.15}$$

取 $z=e^{j\omega}$，式(5.14)可写为

$$\hat{X}(e^{j\omega})=\ln[X(e^{j\omega})]=\ln|X(e^{j\omega})|+j\arg[X(e^{j\omega})] \tag{5.16}$$

式(5.15)可写为

$$\hat{x}(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\hat{X}(e^{j\omega})e^{j\omega n}\,d\omega \tag{5.17}$$

则式(5.17)即为信号 $x(n)$ 的复倒谱 $\hat{x}(n)$ 的定义。在英语中，倒谱 Cepstrum 是将谱 Spectrum 中前四个字母倒置后得到的，因为 $\hat{X}(e^{j\omega})$ 一般为复数，故称 $\hat{x}(n)$ 为复倒谱。如果对 $X(e^{j\omega})$ 的绝对值取对数，得

$$\hat{X}(e^{j\omega})=\ln|X(e^{j\omega})|$$

则 $\hat{X}(e^{j\omega})$ 为实数，由此求出的倒频谱 $c(n)$ 为实倒谱，简称为倒谱，即

$$c(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\ln|X(e^{j\omega})|e^{j\omega n}\,d\omega \tag{5.18}$$

在式(5.16)中，实部是可以取唯一值的，但对于虚部，会引起唯一性问题，因此要求相角为 ω 的连续奇函数。

5.4.2 复倒谱的性质

为判断复倒谱的性质，研究有理 z 变换的一般形式即可。z 变换的一般形式为

$$X(z)=\frac{Az^r\prod_{k=1}^{M_i}(1-a_kz^{-1})\prod_{k=1}^{M_0}(1-b_kz)}{\prod_{k=1}^{N_i}(1-c_kz^{-1})\prod_{k=1}^{N_0}(1-d_kz)} \tag{5.19}$$

其中，a_k、b_k、c_k、d_k 的绝对值皆小于 1；A 是一个非负实系数。因此，$1-a_kz^{-1}$ 和 $1-c_kz^{-1}$ 项对应于单位圆内的零点和极点；$1-b_kz$ 和 $1-d_kz$ 项对应于单位圆外的零点和极点；M_i 和 M_0 分别表示单位圆内和单位圆外的零点数目；N_i 和 N_0 分别表示单位圆内和单位圆外的极点数目；因子 z^r 简单地表示时间原点的移动。于是，$X(z)$ 的复对数为

$$\hat{X}(z)=\ln[A]+\ln[z^r]+\sum_{k=1}^{M_i}\ln(1-a_kz^{-1})+\sum_{k=1}^{M_0}\ln(1-b_kz)$$

$$-\sum_{k=1}^{N_i}\ln(1-c_k z^{-1})-\sum_{k=1}^{N_0}\ln(1-d_k z) \tag{5.20}$$

当在单位圆上估计式(5.20)时，可以看到其中 $\ln[z^r]$ 这一项只在复对数的虚部中出现。它只携带关于时间原点的信息，在计算复倒谱的过程中一般要去掉它，因此，在讨论复倒谱的性质时将这一项略去。每个对数项都可以写成一个幂级数展开式，可以证明复倒谱具有如下形式：

$$\hat{x}(n)=\begin{cases}\ln[A], & n=0\\ \sum\limits_{k=1}^{N_i}\dfrac{c_k^n}{n}-\sum\limits_{k=1}^{M_i}\dfrac{a_k^n}{n}, & n>0\\ \sum\limits_{k=1}^{M_0}\dfrac{b_k^{-n}}{n}-\sum\limits_{k=1}^{N_0}\dfrac{d_k^{-n}}{n}, & n<0\end{cases} \tag{5.21}$$

上式表明了复倒谱的许多重要性质。

性质 1：即使 $x(n)$ 可以满足因果性、稳定性、甚至持续期有限的条件，一般而言复倒谱也是非零的，而且在正负 n 两个方向上都是无限伸展的。

性质 2：复倒谱是一个有界衰减序列，其界限为

$$|\hat{x}(n)|<\beta\frac{\alpha^{|n|}}{|n|},\quad |n|\to\infty \tag{5.22}$$

其中，α 是 a_k、b_k、c_k、d_k 的最大绝对值，而 β 是一个常数。

性质 3：如果 $X(z)$ 在单位圆外无极点和零点(即 $b_k=d_k=0$)，则有

$$\hat{x}(n)=0\quad n<0 \tag{5.23}$$

这种信号称为“最小相位”信号，对于用式(5.23)所表示的序列，有一个通用的结论：这种序列完全可以用它们的傅里叶变换的实部来表示。因此，我们可以单独用傅里叶变换的模的对数值来求最小相位信号的复倒谱。我们知道一个序列的傅里叶变换的实部就等于该序列偶部的傅里叶变换，因为 $\ln|X(e^{j\omega})|$ 是倒频谱 $c(n)$ 的傅里叶变换，所以

$$c(n)=\frac{\hat{x}(n)+\hat{x}(-n)}{2} \tag{5.24}$$

用式(5.23)和式(5.24)，容易证明

$$\hat{x}(n)=\begin{cases}0, & n<0\\ c(n), & n=0\\ 2c(n), & n>0\end{cases} \tag{5.25}$$

因此，为了求得最小相位序列的复倒谱，可以先计算其倒谱 $c(n)$，然后用式(5.25)求 $\hat{x}(n)$。对于最小相位序列的另一个重要结论是复倒谱可由输入信号经过递推计算得到，递推公式是

$$\hat{x}(n)=\begin{cases}\ln[x(0)], & n=0\\ \dfrac{x(n)}{x(0)}-\sum\limits_{k=0}^{n-1}\left(\dfrac{k}{n}\right)\hat{x}(k)\dfrac{x(n-k)}{x(0)}, & n>0\\ 0, & n<0\end{cases} \tag{5.26}$$

性质 4：对于 $X(z)$ 在单位圆内没有极点或零点的情形，可以得到与此类似的结论。这种信号称为“最大相位”信号，在此情况下有

$$\hat{x}(n)=0,\quad n>0 \tag{5.27}$$

如果再一起考虑式(5.24)与式(5.27)，可得

$$\hat{x}(n)=\begin{cases}0, & n>0\\ c(n), & n=0\\ 2c(n), & n<0\end{cases} \tag{5.28}$$

和最小相位序列的情形相同，也能得到一个复倒谱的递推公式，其形式为

$$\hat{x}(n)=\begin{cases}\ln[x(0)], & n=0\\ \dfrac{x(n)}{x(0)}-\sum\limits_{k=n+1}^{0}\left(\dfrac{k}{n}\right)\hat{x}(k)\dfrac{x(n-k)}{x(0)}, & n<0\\ 0, & n>0\end{cases} \tag{5.29}$$

性质 5：如果输入信号为一串冲激信号，它具有如下形式：

$$p(n)=\sum_{r=0}^{M}\alpha_r\delta(n-rN_p) \tag{5.30}$$

式(5.30)的 z 变换是

$$P(z)=\sum_{r=0}^{M}\alpha_r z^{-rN_p} \tag{5.31}$$

由式(5.31)可见，$P(z)$是变量 z^{-N_p} 的多项式而不是 z^{-1} 的多项式，这样，$P(z)$可以表示成若干形式为 $1-az^{-N_p}$ 和 $1-bz^{N_p}$ 的因式的乘积，因而容易看到，复倒谱 $\hat{p}(n)$只在 N_p 的各整数倍点上不为零，这意味着 $\hat{p}(n)$也是一个间隔为 N_p 的冲激串。

例如，设 $p(n)$为

$$p(n)=\delta(n)+\alpha\delta(n-N_p) \tag{5.32}$$

其中 $0<\alpha<1$，则

$$P(z)=1+\alpha z^{-N_p} \tag{5.33}$$

$$\hat{P}(z)=\ln(1+\alpha z^{-N_p})=\sum_{n=1}^{\infty}(-1)^{n+1}\frac{\alpha^n}{n}z^{-nN_p} \tag{5.34}$$

这表明 $\hat{p}(n)$是一个冲激串，冲激之间的间隔为 N_p，即有

$$\hat{p}(n)=\sum_{r=1}^{\infty}(-1)^{r+1}\frac{\alpha^r}{r}\delta(n-rN_p) \tag{5.35}$$

这表明对于一串间隔均匀的冲激，它的复倒谱也是一串均匀间隔的冲激，而且其间隔相同，这对于语音分析是一个很重要的结果。

5.5 复倒谱的几种计算方法

在复倒谱分析中，z 变换后得到的是复数，所以取对数时要进行复对数运算。这时存在相位的多值性问题，称为“相位卷绕”。由于相位卷绕使后面求复倒谱、以及由复倒谱恢复语音等运算均存在不确定性而产生错误。

设信号为

$$x(n)=x_1(n)*x_2(n) \tag{5.36}$$

则其傅里叶变换为

$$X(e^{j\omega})=X_1(e^{j\omega})\cdot X_2(e^{j\omega}) \tag{5.37}$$

对上式取复对数为

$$\ln X(e^{j\omega})=\ln X_1(e^{j\omega})+\ln X_2(e^{j\omega}) \tag{5.38}$$

则其幅度和相位分别为

$$\ln|X(e^{j\omega})|=\ln|X_1(e^{j\omega})|+\ln|X_2(e^{j\omega})| \tag{5.39}$$

$$\varphi(\omega)=\varphi_1(\omega)+\varphi_2(\omega) \tag{5.40}$$

式中，虽然 $\varphi_1(\omega)$，$\varphi_2(\omega)$的范围均在$(-\pi,\pi)$之内，但 $\varphi(\omega)$的值可能超过$(-\pi,\pi)$范围。计算机处理时总相位值只能用其主值 $\Phi(\omega)$表示，然后把这个相位主值“展开”，得到连续相位。所以存在下面的情况：

$$\varphi(\omega)=\Phi(\omega)+2k\pi \quad (k\text{ 为整数}) \tag{5.41}$$

此时即产生了相位卷绕。下面介绍几种避免相位卷绕求复倒谱的方法。

5.5.1 最小相位信号法

这是解决相位卷绕的一种比较好的方法。但它有一个限制条件：即被处理的信号 $x(n)$必须是最小相位信号。实际上许多信号就是最小相位信号，或可以看做最小相位信号。语音信号的模型就是极点都在 z 平面单位圆内的全极点模型，或者极零点都在 z 平面单位圆内的极零点模型。

最小相位信号法是由最小相位信号序列的复倒谱性质及 Hilbert 变换的性质推导出来的。设信号 $x(n)$的 z 变换为 $X(z)=N(z)/D(z)$，则有

$$\hat{X}(z)=\ln X(z)=\ln\frac{N(z)}{D(z)} \tag{5.42}$$

根据 z 变换的微分特性有

$$\begin{aligned}\sum_{n=-\infty}^{\infty}n\hat{x}(n)z^{-n}&=-z\frac{\mathrm{d}}{\mathrm{d}z}\hat{X}(z)=-z\frac{\mathrm{d}}{\mathrm{d}z}\left[\ln\frac{N(z)}{D(z)}\right]=\frac{-z\dfrac{\mathrm{d}}{\mathrm{d}z}\left[\dfrac{N(z)}{D(z)}\right]}{\dfrac{N(z)}{D(z)}}\\&=\frac{\dfrac{-z[D(z)N'(z)-N(z)D'(z)]}{D^2(z)}}{\dfrac{N(z)}{D(z)}}=-z\frac{[D(z)N'(z)-N(z)D'(z)]}{N(z)D(z)}\end{aligned} \tag{5.43}$$

如果 $x(n)$是最小相位信号，则 $N(z)$和 $D(z)$的所有根均在 z 平面的单位圆内；同时，由上式可知，此时 $nx(n)$的 z 变换的所有极点[即上式分母 $N(z)D(z)$的根]也均位于 z 平面的单位圆内。这表明，若 $x(n)$是最小相位信号，则$\hat{x}(n)$必然是稳定的因果序列。

另外，由 Hilbert 变换的性质可知，任一因果的复倒谱序列$\hat{x}(n)$都可以分解为偶对称分量$\hat{x}_e(n)$和奇对称分量$\hat{x}_o(n)$之和，即

$$\hat{x}(n)=\hat{x}_e(n)+\hat{x}_o(n) \tag{5.44}$$

其中

$$\begin{aligned}\hat{x}_e(n)&=[\hat{x}(n)+\hat{x}(-n)]/2\\\hat{x}_o(n)&=[\hat{x}(n)-\hat{x}(-n)]/2\end{aligned} \tag{5.45}$$

而且，这两个分量的傅里叶变换分别为$\hat{x}(n)$的傅里叶变换的实部和虚部。

$$\hat{X}(e^{j\omega})=\sum_{n=-\infty}^{\infty}\hat{x}(n)e^{-jn\omega}=\hat{X}_R(e^{j\omega})+j\hat{X}_I(e^{j\omega}) \tag{5.46}$$

则

$$\hat{X}_{\mathrm{R}}(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{n=-\infty}^{\infty} \hat{x}_{\mathrm{e}}(n)\mathrm{e}^{-\mathrm{j}n\omega} \tag{5.47}$$

$$\hat{X}_{\mathrm{I}}(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{n=-\infty}^{\infty} \hat{x}_{\mathrm{o}}(n)\mathrm{e}^{-\mathrm{j}n\omega} \tag{5.48}$$

由式(5.46)可得

$$\hat{x}(n)=\begin{cases}0, & n<0\\ \hat{x}_{\mathrm{e}}(n), & n=0\\ 2\,\hat{x}_{\mathrm{e}}(n), & n>0\end{cases} \tag{5.49}$$

此即复倒谱的性质3,也就是说一个因果序列可由其偶对称分量来恢复。如果引入一个辅助因子 $g(n)$,式(5.49)可写为

$$\hat{x}(n)=g(n)\cdot\hat{x}_{\mathrm{e}}(n) \tag{5.50}$$

其中

$$g(n)=\begin{cases}0, & n<0\\ 1, & n=0\\ 2, & n>0\end{cases} \tag{5.51}$$

根据上述原理,可以画出最小相位法求复倒谱的原理框图,如图 5.5 所示。由倒谱 $c(n)$ 的定义,可以看出图中$\hat{x}(n)$的偶对称分量$\hat{x}_{\mathrm{e}}(n)$即为 $c(n)$。

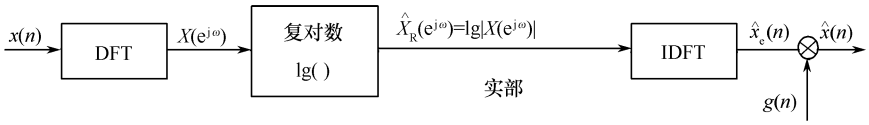


图 5.5　最小相位法求复倒谱的原理框图

5.5.2　递归法

这种方法也仅限于 $x(n)$是最小相位信号的情况。根据 z 变换的微分特性

$$-z\frac{\mathrm{d}}{\mathrm{d}z}\hat{X}(z)=-z\frac{\mathrm{d}}{\mathrm{d}z}[\ln X(z)]=\frac{-z\dfrac{\mathrm{d}X(z)}{\mathrm{d}z}}{X(z)} \tag{5.52}$$

得

$$-zX(z)\frac{\mathrm{d}}{\mathrm{d}z}\hat{X}(z)=-z\frac{\mathrm{d}}{\mathrm{d}z}X(z) \tag{5.53}$$

对上式求逆 z 变换,根据 z 变换的微分特性,有

$$[n\cdot\hat{x}(n)]*x(n)=n\cdot x(n) \tag{5.54}$$

或写为

$$\sum[k\cdot\hat{x}(k)]x(n-k) = nx(n) \tag{5.55}$$

所以

$$x(n) = \sum_{k=-\infty}^{\infty}\left(\frac{k}{n}\right)\hat{x}(k)x(n-k), \quad n\neq 0 \tag{5.56}$$

设 $x(n)$是最小相位序列,而最小相位信号序列一定为因果序列,所以有

$$\begin{cases}x(n)=0, & n<0\\ \hat{x}(n)=0, & n>0\end{cases} \tag{5.57}$$

此时，将式(5.56)写为

$$x(n)=\sum_{k=0}^{n}\left(\frac{k}{n}\right)\hat{x}(k)x(n-k)$$
$$=\sum_{k=0}^{n-1}\left(\frac{k}{n}\right)\hat{x}(k)x(n-k)+\hat{x}(n)x(0) \tag{5.58}$$

上式中，由于$\hat{x}(k)=0(k<0)$及$\hat{x}(n-k)=0(k>n)$，所以求和上下限变为0～n。由上式得递推公式

$$\hat{x}(n)=\frac{x(n)}{x(0)}-\sum_{k=0}^{n-1}\left(\frac{k}{n}\right)\hat{x}(k)\frac{x(n-k)}{x(0)},\quad n>0 \tag{5.59}$$

为此在第一次递归之前应先求出$\hat{x}(0)$，然后进行递推运算，由复倒谱定义

$$\hat{x}(n)=z^{-1}\{\ln z[x(n)]\}=z^{-1}\left\{\ln\left[\sum_{n=-\infty}^{\infty}x(n)z^{-n}\right]\right\} \tag{5.60}$$

可知

$$x(0)=z^{-1}[\ln z[x(0)]]=\ln x(0)\delta(n)=\ln x(0)$$

如果$x(n)$是最大相位序列，则式(5.60)变为

$$g(n)=\begin{cases}0, & n>0\\ 1, & n=0\\ 2, & n<0\end{cases} \tag{5.61}$$

$$\hat{x}(n)=\frac{x(n)}{x(0)}-\sum_{k=n+1}^{0}\left(\frac{k}{n}\right)\hat{x}(k)\frac{x(n-k)}{x(0)},\quad n<0 \tag{5.62}$$

其中，$\hat{x}(0)=\ln x(0)$。

5.5.3 倒谱的 MATLAB 实现

本实验所用的语音样本是用 Cooledit 在普通室内环境下录制的女声“我到北京去”，采样频率为 8kHz，单声道。

【程序 5.1】 cepstrum.m

```
clear all;
                                                    % 倒谱
[s,fs,nbit]=wavread('beijing.wav');                 % 读入一段语音
b=s';                                               % 将 s 转置
x=b(5000:5399);                                     % 取 400 点语音
N=length(x);                                        % 读入语音的长度
S=fft(x);                                           % 对 x 进行傅里叶变换
Sa=log(abs(S));                                     % log 为以 e 为底的对数
sa=ifft(Sa);                                        % 对 Sa 进行傅里叶逆变换
ylen=length(sa);
for i=1:ylen/2
    sa1(i)=sa(ylen/2+1-i);
end
for i=(ylen/2+1):ylen
    sa1(i)=sa(i+1-ylen/2)
end
```

```
% 绘图
figure(1);
subplot(2,1,1);
plot(x);
axis([0,400,－0.5,0.5])
title('截取的语音段');
xlabel('样点数');
ylabel('幅度');
subplot(2,1,2);
time2＝[－199:1:－1,0:1:200];
plot(time2,sa1);
axis([－200,200,－0.5,0.5])
title('截取语音的倒谱');
xlabel('样点数');
ylabel('幅度');
```

倒谱程序运行结果如图 5.6 所示。

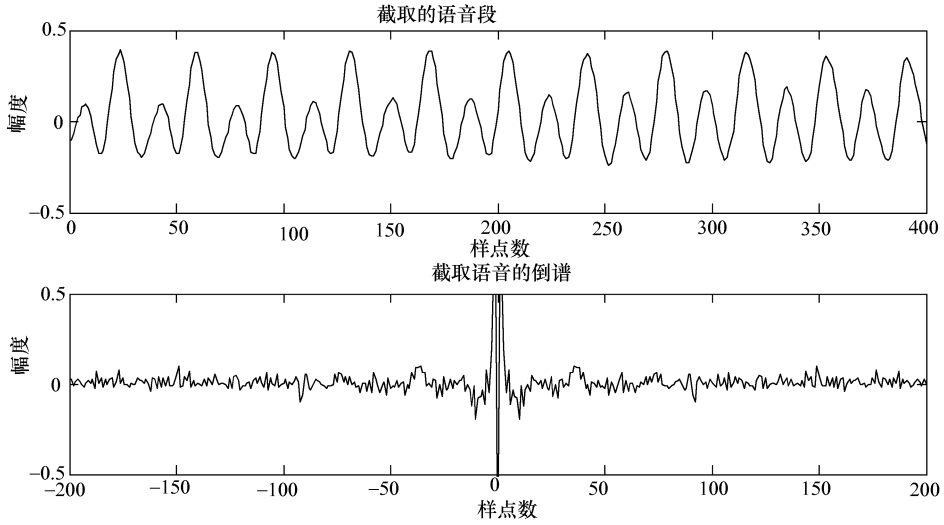


图 5.6　倒谱程序运行结果

5.6　语音的倒谱分析及应用

5.6.1　语音的倒谱分析原理

语音是声道受到激励后产生的，在保持发音器官的位置和形状不变的一段时间内，声道可看做一个线性非时变系统，该系统的输出是声道的冲激响应和激励信号的卷积。因此，可以把浊音语音段看做一个周期冲激串激励一个线性非时变系统产生的。同样，清音语音段可看做由随机噪声激励一个线性非时变系统产生的。于是，浊音语音段可以表示为

$$s(n)=p(n)*g(n)*v(n)*r(n)=p(n)*h_v(n) \tag{5.63}$$

式中，$p(n)$是一个周期冲激串，其周期为 N_p，N_p 即基音周期，$g(n)$为声门波表示式，$v(n)$为声道冲激响应，$r(n)$为辐射冲激响应。而 $h_v(n)$是一个线性系统的冲激响应。

由于

$$p(n) = \sum_{r=0}^{M-1} \delta(n - rN_p) \tag{5.64}$$

故

$$s(n) = \sum_{r=0}^{M-1} h_v(n - rN_p) \tag{5.65}$$

$h_v(n)$可表示为声门波 $g(n)$，声道冲激响应 $v(n)$及辐射冲激响应 $r(n)$的卷积，即

$$h_v(n) = g(n) * v(n) * r(n) \tag{5.66}$$

其 z 变换为

$$H_v(z) = G(z) \cdot V(z) \cdot R(z) \tag{5.67}$$

而对于清音，它的一段可以表示为

$$s(n) = u(n) * v(n) * r(n) = u(n) * h_u(n) \tag{5.68}$$

式中，$u(n)$是一个随机噪声序列，而 $h_u(n)$是一个系统的冲激响应，该系统是声道冲激响应 $v(n)$与辐射响应 $r(n)$的卷积，即

$$h_u(n) = v(n) * r(n) \tag{5.69}$$

其 z 变换为

$$H_u(z) = V(z) \cdot R(z) \tag{5.70}$$

在许多应用中，语音分析的根本任务是解卷积，即求出反映声道系统特性的 $h_v(n)$或 $h_u(n)$，同态解卷系统是解决这一问题的有力手段。

根据语音产生模型，$H_v(z)$(或 $H_u(z)$)总可以用式(5.19)那样的有理分式表示，其复倒谱 $\hat{h}_v(n)$(或$\hat{h}_u(n)$)具有式(5.21)所示的形式，注意到$|a_k|$，$|b_k|$，$|c_k|$，$|d_k|$都小于 1，那么不难看出，$\hat{h}_v(n)$(或$\hat{h}_u(n)$)的绝对值是随 n 的增大而迅速地衰减的。它比相应的 $h_v(n)$或 $h_u(n)$衰减得更快，即更加集中在低时域区。对于浊音来说，它的激励脉冲串在时域和复倒谱域都是间隔为 N_p 的周期性冲激串。在时域的冲激串与 $h_v(n)$是相卷积的关系，各周期之间常常存在混叠，无法把 $h_v(n)$从信号 $s(n)$中很好地分离出来。但是，在复倒谱域冲激串与 $h_v(n)$是相加关系，采用宽度小于 N_p 的复倒谱窗，就可以去掉激励脉冲，得到$\hat{h}_v(n)$的良好估值，再把它通过逆特征系统就可求得 $h_v(n)$，实现解卷。因此，这里的倒谱窗可定义为

$$l(n) = \begin{cases} 1, & |n| < n_0 \\ 0, & |n| \geqslant n_0 \end{cases} \tag{5.71}$$

如果要保存激励分量，选择倒谱窗 $l(n)$为

$$l(n) = \begin{cases} 0, & |n| < n_0 \\ 1, & |n| \geqslant n_0 \end{cases} \tag{5.72}$$

其中 $n_0 < N_p$。倒谱窗在对数幅度谱域起平滑作用。

对于清音来说，清音信号的声道幅度响应$|H_u(e^{j\omega})|$比浊音的要显得平坦一些，共振峰不像浊音那么突出。它的对数 $\ln|H_u(e^{j\omega})|$就显得更平坦了。这样，发清音时的声道响应的倒谱将

是集中在时间原点附近的。当然，用上述倒谱窗对清音信号进行平滑，也可以使 $\ln|H_u(e^{j\omega})|$ 变得更加光滑。

在语音识别的特征提取中，常常不用上述矩形倒谱窗来提取反映声道特性的倒谱系数，而是用一种半个正弦波或类似的两头小中间大的倒谱窗来处理，其效果更好一些。这样的加权倒谱窗有多种形式，其中一种典型的形式为

$$l(n)=\begin{cases}|\sin(\pi n/n_0)|, & |n|<n_0\\ 0, & |n|\geqslant n_0\end{cases} \tag{5.73}$$

这样得到的倒谱系数，称为加权倒谱系数。语音识别的大量实践表明，这种加权倒谱系数用做语音特征参数比不加权的效果更好，但其中权值的选择是很重要的。

用同态解卷积来分离语音的各波形分量很有效。但在许多场合下做语音分析时，只要求估计语音参数，而不是去恢复实际分量的波形。例如，可能只要求判断一段特定的语音是浊音还是清音，如是浊音，则进行基音周期估计，如果是清音要估计频谱等。在这种情况下，不必使用复倒谱，而使用倒谱 $c(n)$。倒谱中的低时分量相当于声道系统函数，而高时分量在浊音时是周期性的，在清音时不是周期性的，没有强烈的峰起，因而利用倒谱可以进行清、浊音判别以及估计浊音的基音周期。语音的倒谱分析系统如图 5.7 所示。

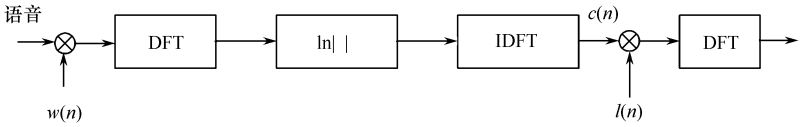


图 5.7　语音的倒谱分析系统

图 5.7 所示的方法已用到语音分析与综合上，根据倒谱的低时部分计算声道冲激响应，还可根据倒谱判别清音或浊音，估计浊音基音周期等，在语音综合时，以声道冲激响应和准周期冲激或噪声序列相卷积来合成语音，也可根据倒谱来估计声道滤波器的极点和零点。语音综合即以二阶时变数字滤波器的级联来实现。利用倒谱分析方法都隐含着声道冲激响应是最小相位的假设。

5.6.2　语音的倒谱应用

1. 基音检测

语音的倒谱是将语音的短时谱取对数后再进行 IDFT 得到的，所以浊音信号的周期性激励反映在倒谱上是同样周期的冲激。借此，可从倒谱波形中估计出基音周期。一般把倒谱波形中第二个冲激，认为是对应激励源的基频。下面给出一种倒谱法求基音周期的框图（如图 5.8所示）及流程图（如图 5.9 所示）。先计算倒谱，然后在预期的基音周期附近寻找峰值。如果倒谱的峰值超出了预先规定的门限，则输入语音段定为浊音，而峰的位置就是基音周期的良好估值。如果没有超出门限的峰值，则输入语音段定为清音。如果计算的是一个时变的倒谱，则可估计出激励源模型及基音周期随时间的变化。一般每隔 10～20ms 计算一次倒谱，这是因为在一般语音中激励参数是缓慢变化的。

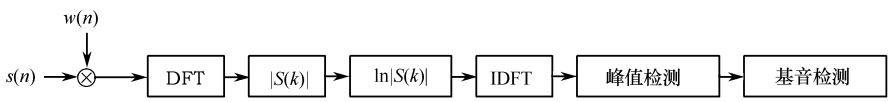


图 5.8　一种倒谱法求基音周期的实现框图

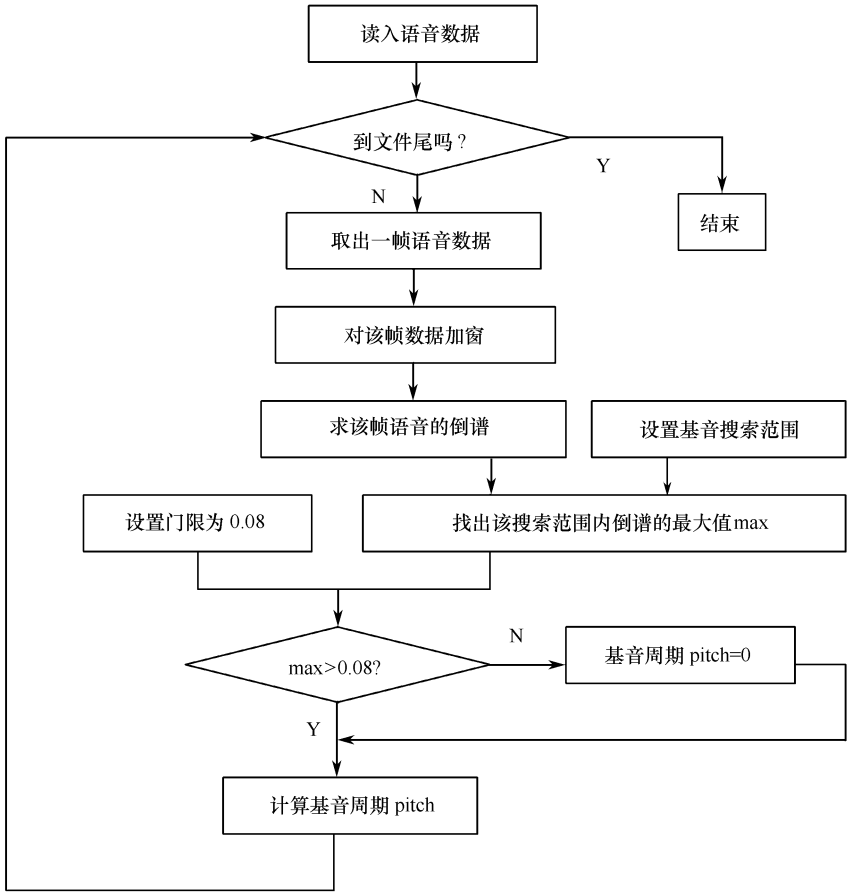


图 5.9　一种倒谱法求基音周期的流程图

【程序 5.2】pitchdetect.m

```
waveFile='beijing.wav';
[y, fs, nbits]=wavread(waveFile);
time1=1:length(y);
time=(1:length(y))/fs;
frameSize=floor(50*fs/1000);                    % 帧长
startIndex=round(5000);                         % 起始序号
endIndex=startIndex+frameSize-1;                % 结束序号
frame=y(startIndex:endIndex);                   % 取出该帧

frameSize=length(frame);
frame2=frame.*hamming(length(frame));           % 加汉明窗
rwy= rceps(frame2);                             % 求倒谱
ylen=length(rwy);
cepstrum=rwy(1:ylen/2);

for i=1:ylen/2
    cepstrum1(i)=rwy(ylen/2+1-i);
```

```
end
for i＝(ylen/2＋1):ylen
    cepstrum1(i)＝rwy(i＋1－ylen/2);
end

% 基音检测
LF＝floor(fs/500);                              % 基音周期的范围是 70～500Hz
HF＝floor(fs/70);
cn＝cepstrum(LF:HF);
[mx_cep ind]＝max(cn);
if mx_cep> 0.08&ind> LF
a＝ fs/(LF＋ind);
else
a＝0;
end
pitch＝a

%  画图
figure(1);
subplot(3,1,1);
plot(time1, y);
title('语音波形');
axis tight
ylim＝get(gca, 'ylim');
line([time1(startIndex),time1(startIndex)],ylim,'color','r');
line([time1(endIndex), time1(endIndex)],ylim,'color','r');
xlabel('样点数');
ylabel('幅度');

subplot(3,1,2);
plot(frame);
axis([0,400,－0.5,0.5])
title('一帧语音');
xlabel('样点数');
ylabel('幅度')

subplot(3,1,3);
time2＝[－199:1:－1,0:1:200];
plot(time2,cepstrum1);
axis([－200,200,－0.5,0.5])
title('一帧语音的倒谱');
xlabel('样点数');
ylabel('幅度');
```

① 浊音：取 startIndex＝round(5000)，其运行结果如图 5.10 所示。

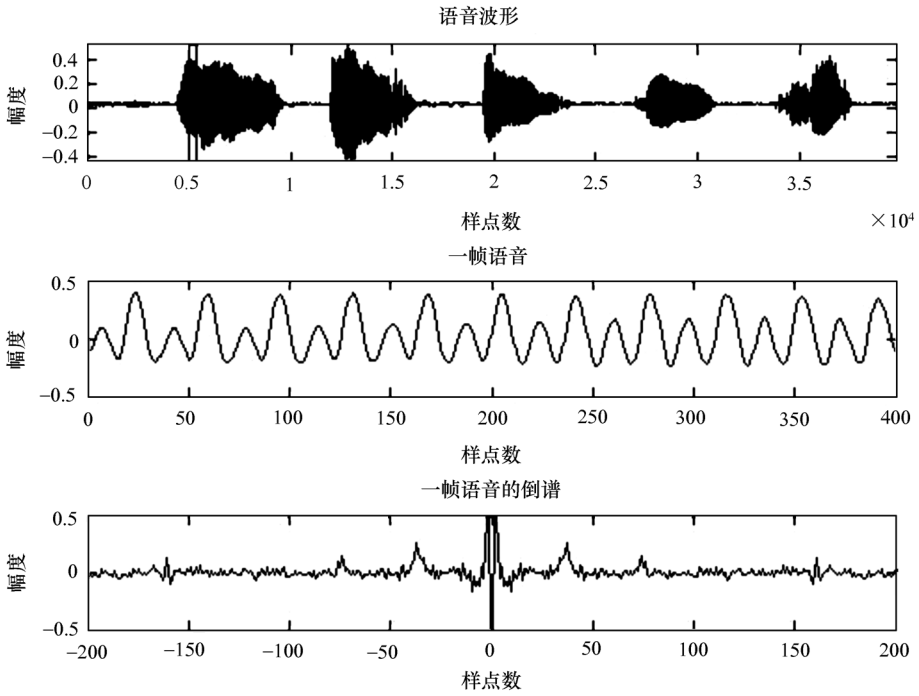


图 5.10　倒谱法求浊音的基音周期

② 清音：取 startIndex＝round(35000)。其运行结果如图 5.11 所示，其中 pitch＝0。

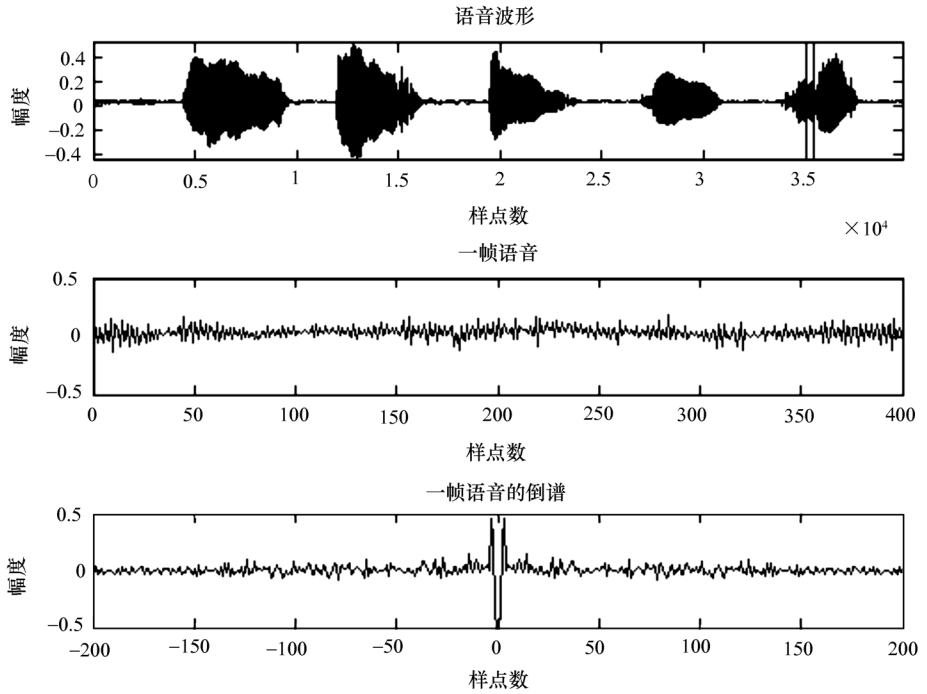


图 5.11　清音的倒谱

2. 共振峰检测

倒谱将基音谐波和声道的频谱包络分离开来。倒谱的低时部分可以分析声道、声门和辐射信息，而高频部分可用来分析激励源信息。对倒谱进行低时窗选，通过语音倒谱分析系统的最后一级，进行 DFT 后的输出即为平滑后的对数模函数，这个平滑的对数谱显示了特定输入语音段的谐振结构，即谱的峰值基本上对应于共振蜂频率，对平滑过的对数谱中的峰值进行定位，即可估计共振峰。原理框图如图 5.12 所示，流程图如图 5.13 所示。

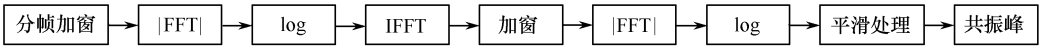


图 5.12　共振峰检测框图

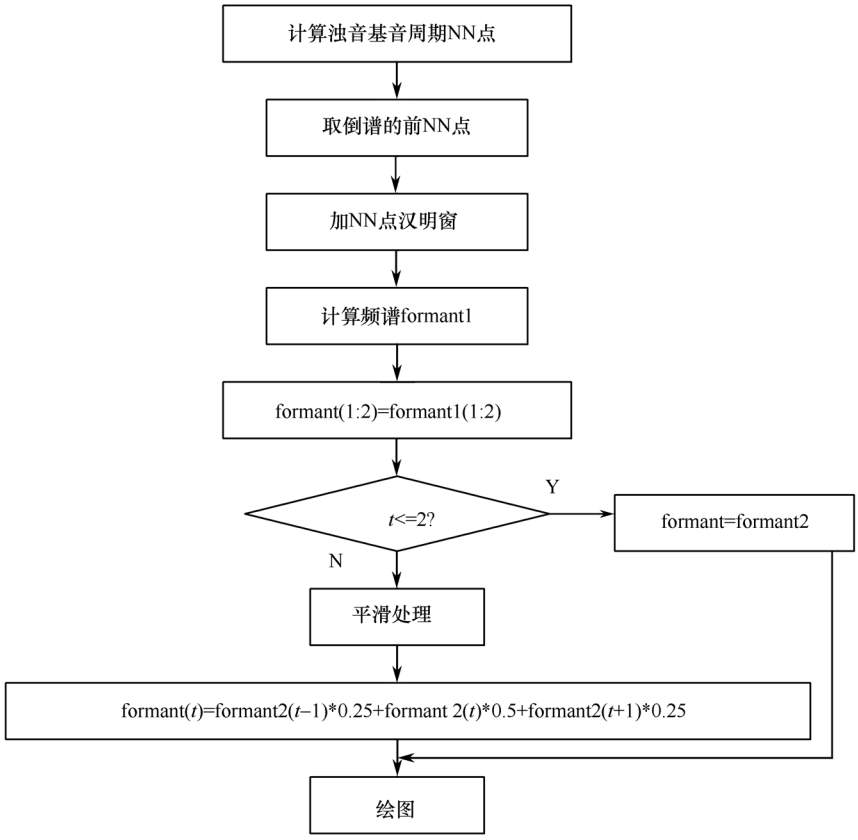


图 5.13　共振峰检测流程图

下面给出共振峰检测的 MATLAB 程序。

【程序 5.3】 formantdetect.m

```
waveFile='qinghua.wav';
[y, fs, nbits]=wavread(waveFile);
time=(1:length(y))/fs;
frameSize=floor(40*fs/1000);              % 帧长
startIndex=round(15000);                  % 起始序号
endIndex=startIndex+frameSize-1;          % 结束序号
frame=y(startIndex:endIndex);             % 取出该帧
```

```
frameSize＝length(frame);
frame2＝frame. * hamming(length(frame));    % 加汉明窗
rwy＝ rceps(frame2);                        % 求倒谱
ylen＝length(rwy);
cepstrum＝rwy(1:ylen/2);

% 基音检测
LF＝floor(fs/500);
HF＝floor(fs/70);
cn＝cepstrum(LF:HF);
[mx_cep ind]＝max(cn);

% 共振峰检测核心代码：
% 找到最大的突起的位置
NN＝ind＋LF;
ham＝ hamming (NN);
cep＝cepstrum(1:NN);
ceps＝cep. * ham;                           % 汉明窗
formant1＝20 * log(abs(fft(ceps)));
formant(1:2)＝formant1(1:2);
for t＝3:NN
% --do some median filtering
    z＝formant1(t－2:t);
    md＝median(z);
    formant2(t)＝md;
end
for t＝1:NN－1
    if t< ＝2
       formant(t)＝formant1(t);
    else
       formant(t)＝formant2(t－1) * 0. 25＋formant2(t) * 0. 5＋formant2(t＋1) * 0. 25;
    end
end

subplot(3,1,1);
plot(cepstrum);
title('倒谱');
xlabel('样点数');
ylabel('幅度')
axis([0,220,－0. 5,0. 5])

spectral＝20 * log10(abs(fft(frame2)));
subplot(3,1,2);
xj＝(1:length(spectral)/2) * fs/length(spectral);
```

```
plot(xj,spectral(1:length(spectral)/2));
title('频谱');
xlabel('频率/Hz');
ylabel('幅度/dB')
axis([0,5500,－100,50])

subplot(3,1,3);
xi＝(1:NN/2) * fs/NN;
plot(xi,formant(1:NN/2));
title('平滑对数幅度谱');
xlabel('频率/Hz');
ylabel('幅度/dB')
axis([0,5500,－80,0])
```

程序运行结果如图 5.14 所示。

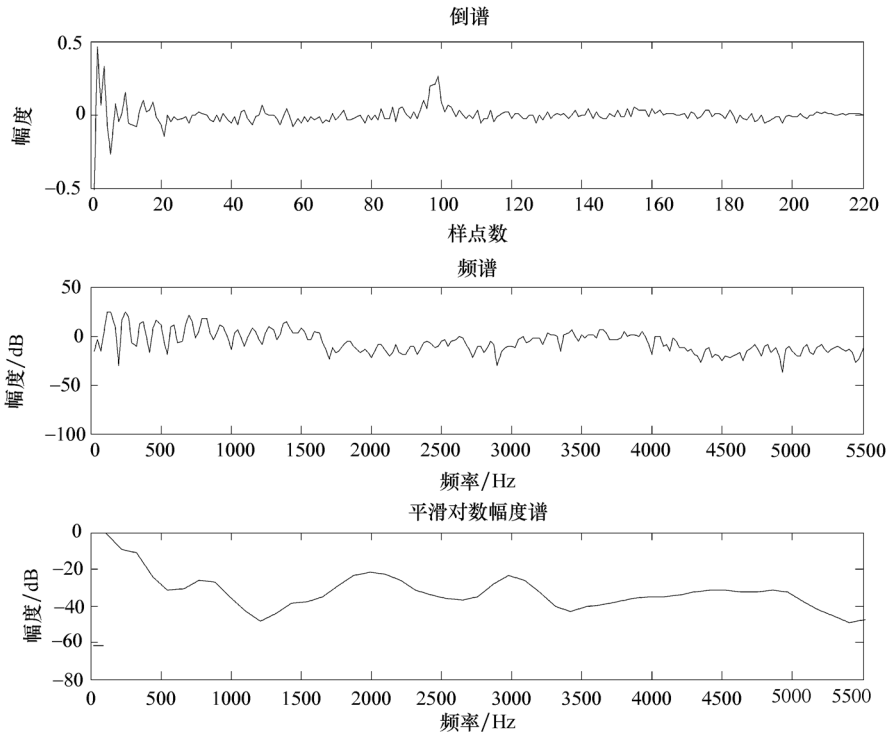


图 5.14　共振峰检测程序运行结果

第 6 章　语音信号线性预测分析

6.1　概　　述

1947 年美国科学家 N. Wiener(维纳)在研究火炮的自动控制时提出了线性预测的思想。1967 年日本学者 Itakura(板仓)等人首先将线性预测技术应用于语音分析和语音合成领域中,使语音处理技术获得巨大的发展。在各种语音处理技术中,线性预测是第一个真正得到实际应用的技术,可用于估计基本的语音参数,如基音周期、共振峰频率、谱特征及声道截面积函数等。

作为最有效的语音分析技术之一,线性预测分析的基本思想是:一个语音取样的现在值可以用若干个语音取样过去值的加权线性组合来逼近。在线性组合中的加权系数称为预测器系数。通过使实际语音抽样和线性预测抽样之间差值的平方和达到最小值,能够决定唯一的一组预测器系数。

线性预测的基本原理是建立在语音的数字模型基础上,为估计数字模型中的参数,线性预测法提供了一种可靠精确而有效的方法。

本章主要介绍语音信号线性预测分析的基本原理,线性预测系数的求解方法及线性预测的几种等价参数。

6.2　LPC 的基本原理

在语音编码算法中,由于实际语音信号的动态变化范围较大,如果直接对其进行量化,则编码所需的比特数较大,编码速率较高。为了保证在较好的语音编码质量前提下,尽量减少编码速率,可设法减小编码器输入信号的动态范围。线性预测编码就是利用过去的样值对新样值进行预测,然后将样值的实际值与其预测值相减得到一个误差信号,显然误差信号的动态范围远小于原始语音信号的动态范围,对误差信号进行量化编码,可大大减少量化所需的比特数,使编码速率降低。

设语音信号的样值序列为 $s(n)$,$n=1,2,\cdots,n$,其中语音信号的当前取样值,即第 n 时刻的取样值 $s(n)$。而 p 阶线性预测,是根据信号过去 p 个取样值的加权和来预测信号当前取样值 $s(n)$,此时的预测器称为 p 阶预测器。设 $\hat{s}(n)$ 为 $s(n)$ 的预测值,则有

$$\hat{s}(n)=\sum_{i=1}^{p}a_i s(n-i) \tag{6.1}$$

式中,$a_1,a_2,\cdots,a_p$ 称为线性预测系数,式(6.1)称为线性预测器,预测器的阶数为 p 阶。p 阶线性预测器的传递函数为

$$P(z)=\sum_{i=1}^{p}a_i z^{-i} \tag{6.2}$$

信号 $s(n)$ 与其线性预测值 $\hat{s}(n)$ 之差称为线性预测误差,用 $e(n)$ 表示。则 $e(n)$ 为

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{6.3}$$

可见，预测误差 $e(n)$ 是信号 $s(n)$ 通过具有如下传递函数的系统输出

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \tag{6.4}$$

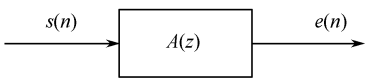


图 6.1　LPC 误差滤波器

如图 6.1 所示。称系统 $A(z)$ 为 LPC 误差滤波器，设计预测误差滤波器 $A(z)$ 就是求解预测系数 $a_1, a_2, \cdots, a_p$，使得预测器的误差 $e(n)$ 在某个预定的准则下最小，这个过程称为 LPC 分析。

线性预测的基本问题就是由语音信号直接求出一组预测系数 $a_1, a_2, \cdots, a_p$，这组预测系数就被看做语音产生模型中系统函数 $H(z)$ 的参数，它使得在一短段语音波形中均方预测误差最小。理论上常用的是均方误差 $E[e^2(n)]$ 最小的准则，$E[\cdot]$ 表示对误差的平方求数学期望或平均值。要得到使 $E[e^2(n)]$ 最小的 a_k，可将 $E[e^2(n)]$ 对各个系数求偏导，并令其结果为零，即

$$\frac{\partial E[e^2(n)]}{\partial a_k} = 2E\left[e(n)\frac{\partial e(n)}{\partial a_k}\right] = 0, \quad k = 1,2,\cdots,p \tag{6.5}$$

由式(6.3)可知

$$\frac{\partial e(n)}{\partial a_k} = -s(n-k), \quad k = 1,2,\cdots,p \tag{6.6}$$

将式(6.6)代入式(6.5)可得

$$-2E[e(n)s(n-k)] = 0, \quad k = 1,2,\cdots,p \tag{6.7}$$

式(6.7)表明预测误差与信号的过去 p 个取样值是正交的，称为正交方程。将式(6.3)代入式(6.7)得

$$E[e(n)s(n-k)] = E\left[s(n)s(n-k) - \sum_{i=1}^{p} a_i s(n-i)s(n-k)\right] = 0, \quad k = 1,2,\cdots,p \tag{6.8}$$

令 $s(n)$ 的自相关序列为

$$R(k) = E[s(n)s(n-k)] \tag{6.9}$$

由于自相关序列为偶对称，因此

$$R(k) = R(-k) = E[s(n)s(n+k)] \tag{6.10}$$

这表明式(6.9)与一般自相关序列的定义是一致的。这样式(6.8)可进一步表示为

$$R(k) - \sum_{i=1}^{p} a_i R(k-i) = 0, \quad k = 1,2,\cdots,p \tag{6.11}$$

式(6.11)称为标准方程式，它表明只要语音信号是已知的，则 p 个预测系数 $a_1, a_2, \cdots, a_p$ 通过求解该方程即可得到。设

$$\boldsymbol{A}_p = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}, \quad \boldsymbol{R}_p = \begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix}, \quad \boldsymbol{R}_p^a = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}$$

式(6.11)矩阵形式为

$$\boldsymbol{R}_p^a - \boldsymbol{R}_p \boldsymbol{A}_p = 0 \quad \text{或} \quad \boldsymbol{A}_p = \boldsymbol{R}_p^{-1} \boldsymbol{R}_p^a \tag{6.12}$$

式中，$\boldsymbol{R}_p^{-1}$ 是 p 阶自相关阵的逆矩阵，通过求解该式即可求得 p 个线性预测系数。

6.3 LPC 和语音信号模型的关系

线性预测分析是建立在语音产生的数字模型基础上的，语音产生的数字模型简化框图如图 6.2 所示。

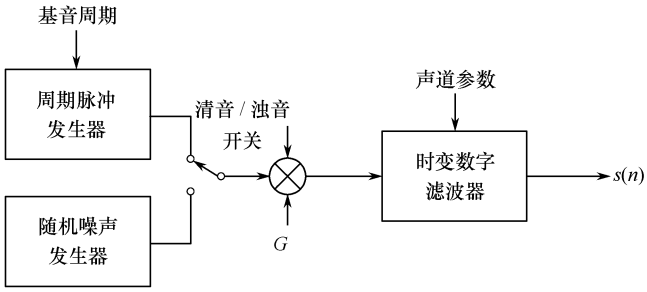


图 6.2　语音产生的数字模型简化框图

该模型的参数有清/浊音判决、浊语音的基音周期、增益常数 G 及数字时变滤波器系数 $a_1, a_2 \cdots, a_p$，这些参数是随时间缓慢变化的。其中，输入的语音信号可由周期脉冲序列的激励(对于浊音)或者随机噪声序列的激励(对于清音)来模拟，周期脉冲序列之间的间隔即为基音周期。而声门激励、声道调制和嘴唇辐射的合成贡献，可用如下数字时变滤波器表示

$$H(z) = \frac{S(z)}{U(z)} = \frac{G\left(1 - \sum_{l=1}^{q} b_l z^{-l}\right)}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{6.13}$$

式(6.13)既有极点又有零点。按其有理式的不同，有如下 3 种信号模型：

① 自回归滑动平均模型(ARMA 模型)。这种模型 $H(z)$既有极点又有零点，是一种一般的模型。此时模型输出 $s(n)$可由信号的过去值 $s(n-i), i=0,1,\cdots,p$ 及输入信号值的线性组合 $u(n-l), l=0,1,\cdots,q$ 来预测得到。

② 自回归信号模型(AR 模型)。此时 $H(z)$只有极点没有零点，模型输出 $s(n)$只由过去的信号值 $s(n-i), i=0,1,\cdots,p$ 线性组合来得到。

③ 滑动平均模型(MA 模型)。此时 $H(z)$只有零点没有极点，模型输出 $s(n)$只由模型的输入 $u(n-l), l=0,1,\cdots,q$ 线性组合来得到。

可见，ARMA 模型是 AR 模型和 MA 模型的混合结构。

声道系统是一个时变系统，但相对于声门激励而言，它是一个随时间 t 而缓慢变化的系统。由声学理论可知，除鼻音和摩擦音时变声道系统 $H(z)$需用零极点模型 ARMA 来模拟外，其他语音均可用全极点 AR 模型来模拟。因为从理论上讲，ARMA 模型和 MA 模型可以用无限高阶的 AR 模型来表示，而且对 AR 模型作参数估计时遇到的是线性方程组的求解问题，处理容易。模型中含有有限个零点时，则需要求解非线性方程组，处理难度大。所以一般都用 AR 模型作为语音信号处理的常用模型。此时时变数字滤波器 $H(z)$写为

$$H(z)=\frac{S(z)}{U(z)}=\frac{G}{1-\sum_{i=1}^{p}a_i z^{-i}} \tag{6.14}$$

式中，增益 G 以及数字滤波器系数 $a_1,a_2,\cdots,a_p$ 都可随时间而变化，p 为预测器阶数。当 p 足够大时，这个全极点模型几乎可以模拟所有语音信号的声道系统。采用这样一个简化模型的主要优点在于可以用线性预测分析法对增益 G 和滤波器系数 $a_1,a_2,\cdots,a_p$ 进行直接而高效的计算。

对图 6.2 的系统，语音抽样信号 $s(n)$ 和激励信号之间的关系可用下列简单的差分方程来表示

$$s(n)=\sum_{i=1}^{p}a_i s(n-i)+Gu(n) \tag{6.15}$$

比较式(6.15)与式(6.3)可以看出，如果语音信号准确服从式(6.15)的模型，则 $e(n)=Gu(n)$，所以预测误差滤波器 $A(z)$ 是式(6.14)中 $H(z)$ 的逆滤波器，故有下式成立

$$H(z)=\frac{G}{A(z)} \tag{6.16}$$

因为图 6.2 所示的模型常用于合成语音，故 $H(z)$ 也称为合成滤波器。而线性预测误差滤波相当于一个逆滤波过程或逆逼近过程，当调整滤波器 $A(z)$ 的参数使输出 $e(n)$ 逼近一个白噪声序列 $u(n)$ 时，$A(z)$ 和 $H(z)$ 是等效的，而按最小均方误差准则求解线性预测系数正是使输出 $e(n)$ 白化的过程。

6.4 LPC 方程的自相关解法及其 MATLAB 实现

根据线性预测分析的原理可知，求解 p 个线性预测系数的依据，是预测误差滤波器的输出方均值或输出功率最小。可称这一最小方均误差为正向预测误差功率 E_p，即

$$\begin{aligned}E_p&=E[e^2(n)]_{\min}=E\{e(n)[s(n)-\sum_{i=1}^{p}a_i s(n-i)]\}\\&=E[e(n)s(n)]-\sum_{i=1}^{p}a_i E[e(n)s(n-i)]\end{aligned} \tag{6.17}$$

由式(6.7)正交方程知，上式第二项为 0。再将式(6.3)代入上式可得

$$E_p=E[e(n)s(n)]=E[s(n)s(n)]-\sum_{i=1}^{p}a_i E[s(n)s(n-i)]=R(0)-\sum_{i=1}^{p}a_i R(i) \tag{6.18}$$

将式(6.18)与式(6.12)组合起来可得

$$\begin{bmatrix}R(0)&R(1)&\cdots&R(p)\\R(1)&R(0)&\cdots&R(p-1)\\R(2)&R(1)&\cdots&R(p-2)\\\vdots&\vdots&&\vdots\\R(p)&R(p-1)&\cdots&R(0)\end{bmatrix}\begin{bmatrix}1\\-a_1\\-a_2\\\vdots\\-a_p\end{bmatrix}=\begin{bmatrix}E_p\\0\\0\\\vdots\\0\end{bmatrix} \tag{6.19}$$

式(6.19)方程的系数矩阵元素是对称的，且沿着任一与主对角线平行的斜对角线上的所有元素相等，系数矩阵大小为 $p\times p$，这样的矩阵称为 Toeplitz(特普利茨)矩阵。式(6.19)称为 Yule-Walker 方程，其中的 $R(p)$ 为根据式(6.9)确定的待分析语音信号 $s(n)$ 的自相关序列。

可见，为了解得线性预测系数，必须首先计算出 $R(k)$，$1\leqslant k\leqslant p$，然后解式(6.19)方程即可。但是计算 $R(k)$，$1\leqslant k\leqslant p$ 却是个十分复杂的问题。为了简化计算，可根据语音信号的短时平稳特性将语音信号分帧，每帧长度取 10～30ms。这样自相关序列 $R(k)$可用下式估计

$$R(k)=E[s(n)s(n-k)]=\frac{1}{n}\sum_{n}s(n)s(n-k) \tag{6.20}$$

如果将预测误差功率 E_p 理解为预测误差的能量，则式(6.20)中的系数$\frac{1}{n}$对式(6.19)方程的求解没有影响，因此可以忽略。但其中的求和范围 n 的不同定义，将会导致不同的线性预测解法。经典的方法有两种：一种是自相关法，该方法假定语音信号序列 $s(n)$在间隔 $0\leqslant n\leqslant N-1$ 以外为 0；这相当于用窗函数从语音序列中截取出选定的序列部分，截取出的序列记为 $s(0),s(1),\cdots,s(N-1)$。另一种是协方差法，该方法不规定语音信号序列 $s(n)$的长度范围，但式(6.20)中 n 的范围为 $0\leqslant n\leqslant N-1$，这样相当于在此范围内估算 $R(k)$所需要的 $s(n)$是存在的。此外协方差法需要确定的是信号序列之间的互相关函数，由此组成的协方差方程组系数矩阵已经不具有 Toeplitz 矩阵的性质，因此其方程的求解不同于自相关法。由于不需要加窗，协方差法计算精度较自相关法大大提高。但由于协方差法不具有自相关法系统稳定性的条件，因此在进行线性预测时，必须随时判定 $H(z)$的极点位置，并加以修正，才能得到稳定的结果。斜格法就是为了解决这两种方法的精度和稳定性之间的矛盾而形成的一种方法。本书只叙述自相关法的详细求解过程，其他两种方法读者可参阅相关书籍。

利用对称 Toeplitz 矩阵的性质，自相关法求解式(6.19)可用 Levinson-Durbin(莱文森－杜宾)递推算法求解。该方法是目前广泛采用的一种方法。算法的计算复杂度为 $O(p^2)$，而线性方程组的一般解法的计算复杂度为 $O(p^3)$，后者比前者要大得多。利用 Levinson-Durbin 算法递推时，从最低阶预测器开始，由低阶到高阶进行逐阶递推计算。其递推过程如下：

$$k_i=\left[r(i)-\sum_{j=1}^{i-1}a_j^{(i-1)}r(i-j)\right]\Big/E_{(i-1)},\quad 1\leqslant i\leqslant p \tag{6.21}$$

$$E_{(0)}=r(0) \tag{6.22}$$

$$E_i=(1-k_i^2)E_{(i-1)} \tag{6.23}$$

$$a_i^{(i)}=k_i \tag{6.24}$$

$$a_j^{(i)}=a_j^{(i-1)}-k_i a_{i-j}^{(i-1)},\quad 1\leqslant j\leqslant i-1 \tag{6.25}$$

式(6.21)至式(6.25)可对 $i=1,2,\cdots,p$ 进行递推求解，其最终解为

$$a_j=a_j^{(p)},\quad 1\leqslant j\leqslant p \tag{6.26}$$

在上面的一组式子中，i 表示预测器阶数，如 $a_j^{(i)}$ 表示 i 阶预测器的第 j 个预测系数。对于 p 阶预测器，在上述求解预测器系数的过程中，阶数低于 p 的各阶预测器系数也同时得到。

自相关法的优点是较简单且结果较稳定，缺点是由于两端的截断效应而精度较低。

程序 6.1 给出用 Levinson-Durbin 递推算法求解线性预测系数的 MATLAB 实现。

【程序 6.1】 lpc_coefficients.m

```
% 此程序的功能是用自相关法求使信号 s 均方预测误差为最小的预测系数
% 算法为 Levinson－Durbin 快速递推算法
% 首先对输入语音进行分帧，并给出 LPC 分析阶次
  fid=fopen('sx86.txt','r');
  p1=fscanf(fid,'% f')
  fclose(fid);
```

```
p2=filter([1 -0.68], 1, p1)                    % 预加重滤波
x=fra(320,160,p2);                             % 将预加重后语音分帧,每帧 320 个样点,
                                               % 帧重叠 160
x=x(60,:);                                     % 取第 60 帧输入信号进行处理,x 为行向量
s=x';                                          % x 为行向量,s 为列向量
N=16;                                          % LPC 阶次 N=16
p=N;                                           % 获得 LPC 阶次
n=length(s);                                   % 获得信号长度

for i=1:p
  Rp(i,1)=sum(s(i+1:n).*s(1:n-i))              % 求向量的自相关函数 .* 表示两个同维
                                               % 矩阵
                                               % 相应元素相乘
  % Rn(i)=sum(s(1:N-i).*s(1+i:N));
end
Rp=Rp(:)                                       % 将自相关函数变为列向量
Rp_0=s'*s;                                     % 即 Rn(0)
Ep=zeros(p,1);                                 % Ep 为 p 阶最佳线性预测反滤波能量
k=zeros(p,1);                                  % k 为自相关系数
a=zeros(p,p);                                  % 以上为初始化
% i=1 的情况需要特殊处理,也就是对 p=1 进行处理
Ep_0=Rp_0;
k(1,1)=Rp(1,1)/Rp_0;
a(1,1)=k(1,1);
Ep(1,1)=(1-k(1,1)^2)*Ep_0;
% i>=2 以后使用递归算法
if p>1
  for i=2:p
      k(i,1)=(Rp(i,1)-sum(a(1:i-1,i-1).*Rp(i-1:-1:1)))/Ep(i-1,1);
%  求式(6.21)k(i)
      a(i,i)=k(i,1);                           % 求式(6.24)a(i)
      Ep(i,1)=(1-k(i,1)^2)*Ep(i-1,1);          % 求式(6.23)Ei
      for j=1:i-1
          a(j,i)=a(j,i-1)-k(i,1)*a(i-j,i-1);
  end                                          % 求式(6.25)a(j,i)
  end
end
c=-a(:,p);                                     % 将 a 矩阵从第 1 到最后一行的第 p 列元
                                               % 素乘以(-1)赋给 c,c 即为最后求得的
                                               % LPC 系数,不包括第一个系数 1
                                               % 得到最终的 LPC 系数 a1,此处 a1 为行
                                               % 向量
a1(1,1)=1.0;                                   % 赋上第一个 LPC 系数 1
for i=2:p+1
  a1(1,i)= c(i-1,1);                           % 得到第 2 到第 p+1 个 LPC 系数
end
```

6.5 模型增益 G 的确定

根据式(6.18)还可求得 E_p 和增益常数 $G=\sqrt{E_p}$。由式(6.14)得

$$Gu(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{6.27}$$

对上式两边乘以 $s(n)$并求平均值,等式右边为

$$\begin{aligned} E[(s(n) - \sum_{i=1}^{p} a_i s(n-i))s(n)] &= E[s^2(n)] - \sum_{i=1}^{p} a_i E[s(n-i)s(n)] \\ &= R(0) - \sum_{i=1}^{p} a_i R(i) \end{aligned} \tag{6.28}$$

等式左边为

$$\begin{aligned} GE[u(n)s(n)] &= E[Gu(n)(Gu(n) + \sum_{i=1}^{p} a_i s(n-i))] \\ &= G^2 E[u^2(n)] + G\sum_{i=1}^{p} a_i E[u(n)s(n-i)] \end{aligned} \tag{6.29}$$

因为假设 $u(n)$为零均值、单位方差的白噪声系列,所以 $E[u^2(n)]=1$,又由于 $u(n)$和 $s(n-i)$不相关,所以 $E[u(n)s(n-i)]=0$,最后得到

$$G^2 = R(0) - \sum_{i=1}^{p} a_i R(i) \tag{6.30}$$

将式(6.30)与式(6.18)比较,可以得出

$$G^2 = E_p \tag{6.31}$$

关于语音数字模型中的激励源有一个问题需要说明。当一个语音信号序列确实是由图 6.2的信号模型产生的,并且激励源是具有平坦谱包络特性的白噪声时(相当于清音语音),应用线性预测误差滤波方法可以求得预测系数和增益,并且 $H(z)$和所分析的语音序列有相同的谱包络特性;但在浊音语音情况下,激励源是一间隔为基音周期的冲激序列,这与线性预测分析中信号源的假设有所不同。但考虑到这样一个事实:$u(n)$是一串冲激组成,意味着大部分时间里它的值是非常小的(零值)。由于采用均方预测误差最小准则来使预测误差 $e(n)$逼近 $u(n)$,和 $u(n)$能量很小这一事实并不矛盾,因此,为简化运算,我们认为,无论是清音还是浊音,图 6.2 的模型都是适合于线性预测分析的。

6.6 线谱对 LSP 分析

在线性预测语音编码中,线性预测合成滤波器 $H(z)=1/A(z)$,其中 $A(z)$为逆滤波器,且 $A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i}$,$a_i, i = 1,2,\cdots,p$ 为线性预测滤波器系数。$H(z)$常被用于重建语音,但当直接对 LPC 系数进行编码时,$H(z)$的稳定性就不能得到保证。由此引出了许多与 LPC 系数等价的表示方法,以用于提高 LPC 系数的鲁棒性,如线谱对 LSP 就是 LPC 系数的一种等价表示形式。LSP 的概念是由 Itakura(板仓)引入的,但是它一直没有被利用,直到后来人们发现利用 LSP 在频域对语音进行频域编码,比其他的变换技术更能改善编码效率,特别是和预测量化方案结合使用的时候。由于 LSP 能够保证线性预测滤波器的稳定性,其小的系数偏差

带来的谱误差也只是局部的，且 LSP 具有良好的量化特性和内插特性，因而已经在许多编码系统中得到成功的应用。LSP 分析的主要缺点是运算量较大。

6.6.1 LSP 的定义和特点

设线性预测逆滤波器 $A(z)$ 为 $A(z)=1-\sum_{i=1}^{p}a_i z^{-i}$。LSP 作为线性预测参数的一种表示形式，可通过求解 $p+1$ 阶对称和反对称多项式的共轭复根得到。其中 $p+1$ 阶对称和反对称多项式表示如下：

$$P(z)=A(z)+z^{-(p+1)}A(z^{-1}) \tag{6.32}$$

$$Q(z)=A(z)-z^{-(p+1)}A(z^{-1}) \tag{6.33}$$

将式(6.32)、式(6.33)中的 $z^{-(p+1)}A(z^{-1})$ 写为

$$z^{-(p+1)}A(z^{-1})=z^{-(p+1)}-a_1 z^{-p}-a_2 z^{-p+1}-\cdots-a_p z^{-1} \tag{6.34}$$

可以推出

$$P(z)=1-(a_1+a_p)z^{-1}-(a_2+a_{p-1})z^{-2}-\cdots-(a_p+a_1)z^{-p}+z^{-(p+1)} \tag{6.35}$$

$$Q(z)=1-(a_1-a_p)z^{-1}-(a_2-a_{p-1})z^{-2}-\cdots-(a_p-a_1)z^{-p}-z^{-(p+1)} \tag{6.36}$$

可见，$P(z)$ 和 $Q(z)$ 分别为对称和反对称的实系数多项式，它们都有共轭复根。可以证明，当 $A(z)$ 的根位于单位圆内时，$P(z)$ 和 $Q(z)$ 的根都位于单位圆上，而且相互交替出现。如果阶数 p 是偶数，则 $P(z)$ 和 $Q(z)$ 各有一个实根，其中 $P(z)$ 有一个根 $z=-1$，$Q(z)$ 有一个根 $z=1$。如果阶数 p 是奇数，则 $Q(z)$ 有 ±1 两个实根，$P(z)$ 没有实根。此处假定 p 是偶数，这样 $P(z)$ 和 $Q(z)$ 各有 $p/2$ 个共轭复根位于单位圆上，共轭复根的形式为 $z_i=\mathrm{e}^{\pm \mathrm{j}\omega_i}$。设 $P(z)$ 的零点为 $\mathrm{e}^{\pm \mathrm{j}\omega_i}$，$Q(z)$ 的零点为 $\mathrm{e}^{\pm \mathrm{j}\theta_i}$，则满足

$$0<\omega_1<\theta_1<\cdots<\omega_{p/2}<\theta_{p/2}<\pi$$

其中，ω_i 和 θ_i 分别为 $P(z)$ 和 $Q(z)$ 的第 i 个根。

$$P(z)=(1+z^{-1})\prod_{i=1}^{p/2}(1-z^{-1}\mathrm{e}^{\mathrm{j}\omega_i})(1-z^{-1}\mathrm{e}^{-\mathrm{j}\omega_i})=(1+z^{-1})\prod_{i=1}^{p/2}(1-2\cos\omega_i z^{-1}+z^{-2}) \tag{6.37}$$

$$Q(z)=(1-z^{-1})\prod_{i=1}^{p/2}(1-z^{-1}\mathrm{e}^{\mathrm{j}\theta_i})(1-z^{-1}\mathrm{e}^{-\mathrm{j}\theta_i})=(1-z^{-1})\prod_{i=1}^{p/2}(1-2\cos\theta_i z^{-1}+z^{-2}) \tag{6.38}$$

其中，$\cos\omega_i$，$\cos\theta_i$，$i=1,2,\cdots,p/2$ 就是 LSP 系数在余弦域的表示，ω_i、θ_i 则是与 LSP 系数对应的线谱频率 LSF。由于 LSP 参数 ω_i 和 θ_i 成对出现，且反映信号的频谱特性，因此称为线谱对。它们就是线谱对分析所要求解的参数。

下面对 LSP 参数的特性归纳如下：

① LSP 参数都在单位圆上且满足降序排列的特性。

② 与 LSP 参数对应的 LSF 都满足升序排列的顺序特性，且 $P(z)$ 和 $Q(z)$ 的根相互交替出现，这可使与 LSP 参数对应的 LPC 滤波器的稳定性得到保证。因为它保证了在单位圆上，任何时候 $P(z)$ 和 $Q(z)$ 不可能同时为零。

③ LSP 参数都具有相对独立的性质，如果某个特定的 LSP 参数中只移动其中任意一个线谱频率 ω_i 的位置，那么它所对应的频谱只在 ω_i 附近与原始语音频谱有差异，而在其他 LSP 频率上则变化很小。这一特性有利于 LSP 参数的量化和内插。在对 LSP 参数进行矢量量化时可以把码本分裂为几个低维矢量分别进行，这样不仅大大减少搜索量、存储量和训练量，又可以使整体质量得以保持。

④ LSP 参数能够反映声道幅度谱的特点，在幅度大的地方分布较密，反之较疏。这样就相当于反映出了幅度谱中的共振峰特性。因为按照线性预测分析的原理，语音信号的谱特性可以由 LPC 模型谱来估计，将式(6.32)、式(6.33)相加可得

$$A(z)=\frac{1}{2}[P(z)+Q(z)] \tag{6.39}$$

这样，功率谱可以表示为

$$\begin{aligned}|H(e^{j\omega})|^2 &= \frac{1}{|A(e^{j\omega})|^2}=4|P(e^{j\omega})+Q(e^{j\omega})|^{-2}\\ &=2^{-p}\left[\sin^2(\omega/2)\prod_{i=1}^{p/2}(\cos\omega-\cos\theta_i)^2+\cos^2(\omega/2)\prod_{i=1}^{p/2}(\cos\omega-\cos\omega_i)^2\right]^{-1}\end{aligned} \tag{6.40}$$

在式(6.40)中，当 ω 接近于 0 或者 θ_i，$i=1,2,\cdots,p/2$ 时，中括号中的第一项接近于零，当 ω 接近于 π 或者 ω_i，$i=1,2,\cdots,p/2$ 时，中括号中的第二项接近于零，如果 ω_i($i=1,2,\cdots,p/2$)与 θ_i，$i=1,2,\cdots,p/2$ 之间很靠近，则当 ω 接近这些频率时，$|A(j\omega)|^2$ 变小，$|H(j\omega)|^2$ 显示出强谐振特性，相应地语音信号谱包络在这些频率处出现峰值，因此可以说，LSP 分析是用 p 个离散频率 ω_i、θ_i($i=1,2,\cdots,p/2$)的分布密度来表示语音信号谱特性的一种方法。即在语音信号幅度谱较大的地方 LSP 的分布较密，反之较疏。

⑤ 相邻帧 LSP 参数之间都具有较强的相关性，便于语音编码时帧间参数的内插。

图 6.3 为 $p=16$ 时，16 阶 LPC 系数构成的 17 阶对称和反对称多项式 $P(z)$和 $Q(z)$的根在单位圆上的分布图。其中“×”为 $Q(z)$的根所在位置，O 为 $P(z)$的根在单位圆上所在位置。可见，$P(z)$和 $Q(z)$的根在单位圆上是交替出现的。

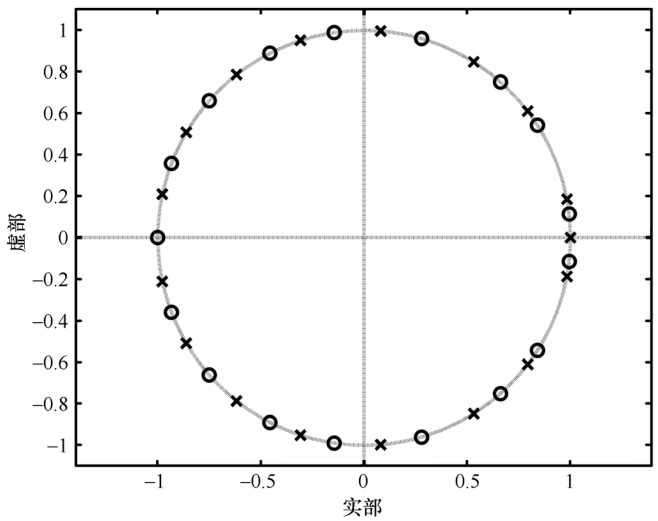
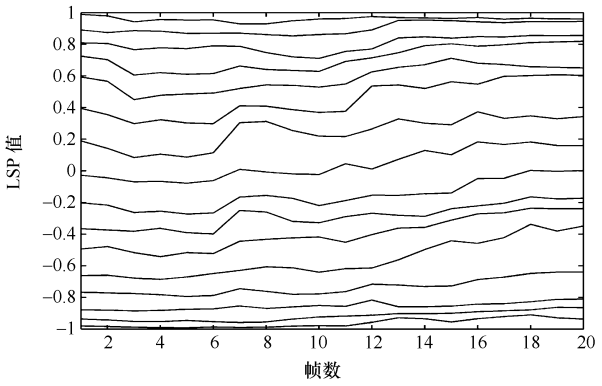
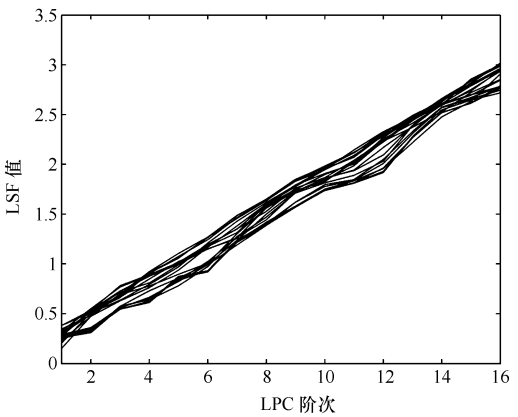


图 6.3　$P(z)$和 $Q(z)$的根在单位圆上的分布图

图 6.4 给出了连续 20 帧 16 阶语音信号的 LSP 轨迹图及 LSF 图，其中图 6.4(a)为连续 20 帧语音信号的 LSP 轨迹图，图中从上往下的曲线分别为 $\cos\omega_i$、$\cos\theta_i$，$i=1,2,\cdots,p/2$，两者交错出现。图中的曲线较好的反应了 LSP 参数的顺序特性，表明了每一帧的 16 个 LSP 参数满足降序排列的特性，帧间的同一个 LSP 参数则比较接近。图 6.4(b)为连续 20 帧语音信号的 LSF 图，可以看出，所有 LSF 曲线都是升序排列，且各帧的同一个 LSF 参数都比较接近。



(a) 连续 20 帧语音信号的 LSP 轨迹图



(b) 连续 20 帧语音信号的 LSF 图

图 6.4　连续 20 帧语音信号的 LSP 轨迹图与 LSF 图

图 6.5 给出了一帧语音信号的 16 阶 LPC 谱包络和相应的 LSF(归一化频率 0～π)，其中实垂线所确定的频率 $f_1, f_3, \cdots, f_{15}$ 与 $P(z)$ 的根 $e^{j\omega_i}$ 的频率对应，虚垂线所确定的频率 $f_2, f_4, \cdots, f_{16}$ 与 $Q(z)$ 的根 $e^{j\theta_i}$ 的频率对应，二者相互交错出现。可以看出，在 LPC 谱包络共振峰区域，LSF 的分布较密，谱谷区域则分布较疏。

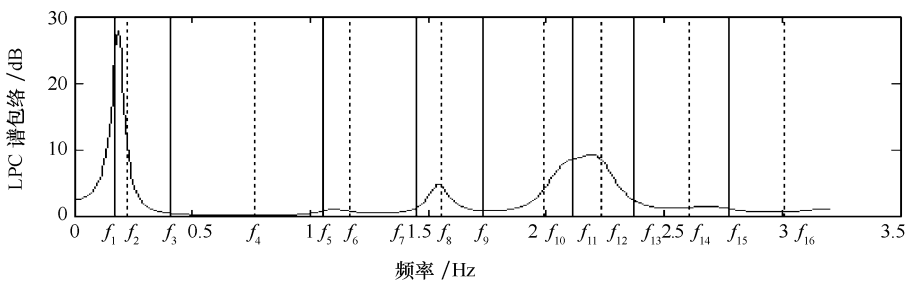


图 6.5　一帧语音信号的 16 阶 LPC 谱包络和相应的 LSF

下面给出绘制图 6.4 所用的 MATLAB 程序。

【程序 6.2】a_lsf_continue_20frame.m

```
% a_lsf_continue_20frame.m
clear;close all
clc
fid=fopen('sx86.txt','r');
p1=fscanf(fid,'% f')
fclose(fid);
p=filter([1 -0.68], 1, p1)% 预加重滤波
x1=fra(320,160,p)% 分帧,每帧 320 个样点,帧重叠 160 个样点
for i=60:79 % 取出第 60 到 79 帧的信号进行分析
x=x1(i,:)
a1=lpc(x,16)
a=a1(:);% 将线性预测系数赋给矩阵 a
lsf=a_lsf_conversion(a) % 调用函数 a_lsf_conversion 实现从 LPC 系数到 lsf 参数的转
                        % 换,函数 a_lsf_conversion()见 6.6.2 节所赋程序。
lsp=cos(lsf)
hold on % 让连续 20 帧 lsp 绘制在一个图形 figure(1)中
figure(1);
        for j=1:16
        lsp1(i-59,j)=lsp(j);
        end
figure(2);
plot(lsf)
end
% EOF a_lsf_continue_20frame.m
```

上述程序运行后,即可在 figure(2)中得到图 6.4(b),绘制图 6.4(a)连续 20 帧 lsp 轨迹的方法如下:程序运行后,打开 MATLAB 中的 Workspace,单击打开 lsp1 矩阵,其为一个 20×16 的矩阵,以每一列为对象单击 Array Editor 中  图标的 plot 按钮,并以 figure(1)作为绘图区域即可。

6.6.2 LPC 参数到 LSP 参数的转换及 MATLAB 实现

在进行语音编码时,要对 LPC 参数进行量化和内插,就需要将 LPC 系数转换为 LSP 系数。为计算方便,将式(6.37)、式(6.38)与 LSP 系数无关的两个实根取掉,得到如下两个新的多项式 $P'(z)$和 $Q'(z)$。

$$P'(z)=\frac{P(z)}{1+z^{-1}}=\prod_{i=1}^{p/2}(1-z^{-1}\mathrm{e}^{\mathrm{j}\omega_i})(1-z^{-1}\mathrm{e}^{-\mathrm{j}\omega_i})=\prod_{i=1}^{p/2}(1-2\cos\omega_i z^{-1}+z^{-2}) \quad (6.41)$$

$$Q'(z)=\frac{Q(z)}{1-z^{-1}}=\prod_{i=1}^{p/2}(1-z^{-1}\mathrm{e}^{\mathrm{j}\theta_i})(1-z^{-1}\mathrm{e}^{-\mathrm{j}\theta_i})=\prod_{i=1}^{p/2}(1-2\cos\theta_i z^{-1}+z^{-2}) \quad (6.42)$$

从 LPC 系数到 LSP 系数的转换过程,其实就是求解使式(6.41)、式(6.42)等于零时的 $\cos\omega_i$、$\cos\theta_i$ 的值,可采用以下几种方法求解。

第一种方法是利用代数方程式求解。

在式(6.41)中,等式的右端可进一步表示为

$$\begin{aligned}1-2\cos\omega_i z^{-1}+z^{-2}&=2z^{-1}(0.5z-\cos\omega_i+0.5z^{-1})\\&=2z^{-1}[0.5(z+z^{-1})-\cos\omega_i]\end{aligned}\tag{6.43}$$

令 $z=e^{j\omega}$,则由欧拉公式 $e^{j\omega}=\cos\omega+j\sin\omega$,可得 $z+z^{-1}=2\cos\omega=2x$。因此式(6.41)、式(6.42)就是关于 x 的一对 $p/2$ 次代数方程式,其系数决定于 a_i,$i=1,2,\cdots,p$,且 a_i,$i=1,2,\cdots,p$ 是已知的,可以用牛顿迭代法来求解。

第二种方法是离散傅里叶变换(DFT)方法。

对 $P'(z)$和 $Q'(z)$的系数求离散傅里叶变换,得到 $z_k=\exp\left(-\frac{jk\pi}{N}\right)$,$k=0,1,\cdots,N-1$ 各点的值,搜索最小值的位置,即是零点所在。由于除了 0 和 π 之外,总共有 p 个零点,而且 $P'(z)$和 $Q'(z)$的根是相互交替出现的,因此只要很少的计算量即可解得,其中 N 的取值取 64～128 就可以。

第三种方法是利用切比雪夫(Chebyshev)多项式求解。

用切比雪夫多项式估计 LSP 系数,可直接在余弦域得到。$z=e^{j\omega}$时,$P'(z)$和 $Q'(z)$可以写为

$$P'(z)=2e^{-jp\omega/2}C(x)\tag{6.44}$$

$$Q'(z)=2e^{-jp\theta/2}C(x)\tag{6.45}$$

其中

$$C(x)=T_{\frac{p}{2}}(x)+f(1)T_{\frac{p}{2}-1}(x)+f(2)T_{\frac{p}{2}-2}(x)+\cdots+f(\frac{p}{2}-1)T_1(x)+f\left(\frac{p}{2}\right)/2\tag{6.46}$$

式中,$T_m(x)=\cos mx$ 是 m 阶的 Chebyshev 多项式。$f(i)$是由递推关系计算得到的 $P'(z)$和 $Q'(z)$的每个系数。由于 $P'(z)$和 $Q'(z)$是对称和反对称的,所以每个多项式只计算前 5 个系数即可。用下面的递推关系可得

$$\begin{cases}f_1(i+1)=a_{i+1}+a_{p-i}-f_1(i),\\f_2(i+1)=a_{i+1}-a_{p-i}+f_2(i),\end{cases}\quad i=0,1,\cdots,p/2\tag{6.47}$$

其中 $f_1(0)=f_2(0)=1.0$。多项式 $C(x)$在 $x=\cos\omega$ 时的递推关系是:

$$\text{for}\quad k=\frac{p}{2}-1\quad\text{to}\quad 1$$

$$\lambda_k=2x\lambda_{k+1}-\lambda_{k+2}+f\left(\frac{p}{2}-k\right)$$

$$\text{end}$$

$$C(x)=x\lambda_1-\lambda_2+f\left(\frac{p}{2}\right)/2$$

其中初始值 $\lambda_{\frac{p}{2}}=1$,$\lambda_{\frac{p}{2}+1}=0$。

第四种方法是将 0～π 之间均分为 60 个点,以这 60 个点的频率值代入式(6.41)、式(6.42),检查它们的符号变化,在符号变化的两点之间均分为 4 份,再将这三个点频率值代入方程式(6.41)、式(6.42),符号变化的点即为所求的解。这种方法误差略大,计算量较大,但程序实现容易。

下面给出从 LPC 参数到 LSP 参数转换的 MATLAB 程序,其中 a_lsf_conversion.m 为求解 LSF 的函数,a_lsf_main.m 为主程序。由于 MATLAB 程序本身有求多项式根的函数,因

此在求解 $P'(z)$和 $Q'(z)$零点时直接调用即可，这极大简化了求解过程。如果用 C 语言编程实现，则上述第四种方法由于编程较容易，因此在语音编码标准中用的较多。如在自适应多速率宽带及窄带语音编码标准 AMR-WB、AMR-NB 中以及 G.729 编码标准中，就使用了这种求解方法。

【程序 6.3】a_lsf_main.m

```
% 已知语音文件求出其 LPC 系数后，调用 a_lsf_conversion.m 函数求其对应的 LSF
% a_lsf_main.m
clear;close all% 将所有变量置为 0
clc% 清除命令窗口
fid=fopen('sx86.txt','r');
p1=fscanf(fid,'% f')
fclose(fid);
p=filter([1 -0.68], 1, p)% 预加重滤波
x=fra(320,160,p) % 分帧，帧移为 160 个样点
x=x(60,:)% 取第 60 帧作为分析帧
N=16% 给线性预测分析的阶次赋值
a1=lpc(x,N)% 调用 MATLAB 库函数中的 lpc 函数求解出 LPC 系数 a1
% 此处也可以调用本章赋的函数 lpc_coeffi-
% cients(s,p)，调用语句为 a1= lpc_coefficients(x,N)
a=a1(:);% 将线性预测系数 a1 赋给矩阵 a
lsf=a_lsf_conversion(a) % 调用函数 a_lsf_conversion 实现从 LPC 系数到 LSF 参数的
                        % 转换
% lsf=poly2lsf(a); % 也可调用 MATLAB 库函数中的 poly2lsf(a)函数求解出 LSF 系数，调用
                   % 结果为归一化角频率
lsf_abnormalized=lsf. * (6400/3.14); % 将求得的 lsf 参数反归一化，反归一化到
                                     % 0～6400Hz
% 使用时可根据实际需要进行更改，如窄带语音编码语音信号频带范围为 300～3400Hz，此时就
% 需要将 6400Hz 改为 3400Hz
% 将求得的归一化、反归一化 lsf 参数输出到文本文件：从 lpc 系数解得的 lsf 参数.txt
fid= fopen('从 lpc 系数解得的 lsf 参数.txt','w');
        fprintf(fid,'归一化的 lsf:\n');
        fprintf(fid,'% 6.2f, ',lsf);
        fprintf(fid,'\n');
        fprintf(fid,'反归一化的 lsf:\n');
        fprintf(fid,'% 8.4f, ',lsf_abnormalized);
        fclose(fid);
% EOF a_lsf_main.m
```

函数 a_lsf_conversion 的 MATLAB 程序见 a_lsf_conversion.m。

```
% 程序 a_lsf_conversion.m。
function lsf=a_lsf_conversion(a)
% 如果 a 不是实数，输出错误信息：LSF 不适用于复多项式的求解
if ～isreal(a),
    error('Line spectral frequencies are not defined for complex polynomials.');
```

```
end
% 如果 a(1)不等于 1,将其归一化为 1
if a(1) ~= 1.0,
    a=a./a(1);% 将矩阵 a 的每个元素除以 a(1)再赋给矩阵 a
end
% 判断线性预测多项式的根是否都在单位圆内,如果不在,则输出错误信息
if (max(abs(roots(a))) > = 1.0),
    error('The polynomial must have all roots inside of the unit circle.');
end
% 求对称和反对称多项式的系数
p=length(a)-1;          % 求对称和反对称多项式的阶次
a1=[a;0];               % 给行矩阵 a 再增加一个元素为 0 的行
a2=a1(end:-1:1);        % a2 的第一行为 a1 的最后一行,最后一行为 a1 的第一行
P1=a1+a2;               % 求对称多项式的系数
Q1=a1-a2;               % 求反对称多项式的系数
% 如果阶次 p 为偶数次,从 P1 取掉实数根 z =-1,从 Q1 取掉实数根 z =1
% 如果阶次为奇数次,从 Q1 取掉实数根 z=1 及 z =-1
if rem(p,2), % 求解 p 除以 2 的余数,如果 p 为奇数次,余数为 1,否则为 0
    Q=deconv(Q1,[1 0 -1]);% 奇数阶次,从 Q1 取掉实数根 z=1 及 z =-1
    P=P1;
else          % p 为偶数阶次执行下面操作
    Q=deconv(Q1,[1 -1]);% 从 Q1 取掉实数根 z=1
    P=deconv(P1,[1 1]);% 从 P1 取掉实数根 z =-1
end
rP=roots(P);% 求去掉实根后的多项式 P 的根
rQ=roots(Q);% 求去掉实根后的多项式 Q 的根
aP=angle(rP(1:2:end));% 将多项式 P 的根转换为角度(为归一化角频率)赋给 ap
aQ=angle(rQ(1:2:end));% 将多项式 Q 的根转换为角度(为归一化角频率)赋给 aQ
lsf=sort([aP;aQ]);% 将 P、Q 的根(归一化角频率)按从小到大顺序排序后即为 lsf
% EOF a_lsf_conversion.m
```

6.6.3 LSP 参数到 LPC 参数的转换及 MATLAB 实现

LSP 系数被量化和内插后,(在解码时)应转换回 LPC 系数 a_i,$i=1,2,\cdots,p$。已知量化和内插的 LSP 系数 q_i,$i=0,1\cdots,p-1$,可用式(6.41)、式(6.42)计算 $P'(z)$和 $Q'(z)$的系数$p'(i)$和 $q'(i)$,以下的递推关系可利用 q_i,$i=0,1,\cdots,p-1$,来计算 $p'(i)$:

for $i=1$ to $p/2$

$p'(i)=-2q_{2i-1}p'(i-1)+2p'(i-2)$

for $j=i-1$ to 1

$p'(j)=p'(j)-2q_{2i-1}p'(j-1)+p'(j-2)$

end

end

其中的 $q_{2i-1}=\cos\omega_{2i-1}$,初始值 $p'(0)=1$,$p'(-1)=0$。把上面递推关系中的 q_{2i-1}替换为 q_{2i},就可以得到 $q'(i)$。

一旦得出系数 $p'(i)$和 $q'(i)$，就可以得到 $P'(z)$和 $Q'(z)$，$P'(z)$乘以 $1+z^{-1}$得到 $P(z)$，$Q'(z)$乘以 $1-z^{-1}$得到 $Q(z)$，即

$$\begin{cases} p_1(i)=p'(i)+p'(i-1), & i=1,2,\cdots,p/2 \\ q_1(i)=q'(i)-q'(i-1), & i=1,2,\cdots,p/2 \end{cases} \tag{6.48}$$

最后得到 LPC 系数为

$$a_i=\begin{cases} 0.5p_1(i)+0.5q_1(i), & i=1,2,\cdots,p/2 \\ 0.5p_1(p+1-i)-0.5q_1(p+1-i), & i=p/2+1,p/2+2,\cdots,p \end{cases} \tag{6.49}$$

这是直接从关系式 $A(z)=\frac{1}{2}[P(z)+Q(z)]$得到的，并且考虑了 $P(z)$和 $Q(z)$分别是对称和反对称多项式。

以上从 LSP 参数到 LPC 系数的求解过程，运用 C 语言实现时，读者可参阅语音编码算法标准 AMR-NB 以及 G.729。如果用 MATLAB 实现，则可利用 MATLAB 自带的函数poly( )来得到 $P'(z)$和 $Q'(z)$多项式，下面给出用 MATLAB 实现从 LSF 到 LPC 系数的求解过程。

```
function a=lsf_lpc_conversion(lsf)
% 功能:将线谱频率 LSF 转换为 LPC 系数,其中形参 lsf 为行向量
% lsf_lpc_conversion.m
% 如果线谱频率 lsf 是复数,则返回错误信息
if (~isreal(lsf)),
    error ('Line spectral frequencies must be real.');
end
% 如果线谱频率 lsf 不在 0-pi 范围,则返回错误信息
if (max(lsf) > pi || min(lsf)< 0),
    error ('Line spectral frequencies must be between 0 and pi.');
end

lsf=lsf(:);% 将 lsf 转换为列向量
p=length(lsf); % lsf 阶次为 p
% 用 lsf 形成零点
z= exp(j*lsf);
rP=z(1:2:end);% 把 z(1)、z(3)到 z(p-1)赋给 rP
rQ=z(2:2:end);% 把 z(2)、z(4)到 z(p)赋给 rQ
% 把共轭复根考虑上
rQ=[rQ;conj(rQ)];% 把 rQ 的共轭复根赋上
rP=[rP;conj(rP)];% 把 rP 的共轭复根赋上
% 构成多项式 P 和 Q,注意必须是实系数
Q =poly(rQ);
P =poly(rP);
% 考虑上 z=1 和 z=-1 以形成对称和反对称多项式
if rem(p,2),
    % 如果是奇数阶次,则 z=+1 和 z=-1 都是 Q1(z)的根
    Q1=conv(Q,[1 0 -1]);
    P1=P;
else
```

```
    % 如果是偶数阶次,则 z=-1 是对称多项式 P1(z)的根,z =1 是反对称多项式 Q1(z)的根
    Q1=conv(Q,[1 -1]);
    P1=conv(P,[1 1]);
end
% 由 P1 和 Q1 求解 LPC 系数
a=.5*(P1+Q1);
a(end)=[]; % 最后一个系数是 0,不返回
% [EOF] lsf_lpc_conversion.m
```

调用该函数的调用语句如下：

```
a2=lsf_lpc_conversion(lsf) ; % 调用函数 lsf_lpc_conversion( )实现从 LSF 参数到 LPC
                             % 系数的转换
```

6.7 导抗谱对 ISP 分析

在线性预测语音编码中,为了提高 LPC 系数的鲁棒性,引出了许多与 LPC 系数等价的表示方法,除线谱对 LSP 外,导抗谱对 ISP 也是与 LPC 系数等价的一种参数。在语音编码中,可将 LPC 系数转换为导抗谱对 ISP 以进行量化和内插。导抗谱对 ISP 是由 Yuval Bistritz 和 Shlomo Peller 在 1993 年提出的,是用于提高 LPC 系数鲁棒性的一种等价表示方法,它的提出形成了一种用于体现 LPC 滤波器特性的新参数。目前已经用于自适应多速率宽带(AMR-WB)语音编码算法中。

6.7.1 ISP 的定义和特点

设线性预测逆滤波器 $A(z)$ 为 $A(z)=1-\sum_{i=1}^{p}a_i z^{-i}$。用 LPC 系数 a_i,$i=1,2,\cdots,p$ 构造对称和反对称多项式 $P(z)$、$Q(z)$如下

$$P_1(z)=A(z)+z^{-p}A(z^{-1}) \tag{6.50}$$

$$Q_1(z)=A(z)-z^{-p}A(z^{-1}) \tag{6.51}$$

由此得到一个反映声门激励的导抗函数 $I_p(z)$,其表示式如下

$$I_p(z)=\frac{A(z)-z^{-p}A(z^{-1})}{A(z)+z^{-p}A(z^{-1})}=\frac{Q_1(z)}{P_1(z)} \tag{6.52}$$

ISP 就是由 $I_p(z)$的极点和零点所构成的,另外还包括了一个反射系数 k_p。由于 $I_p(z)$导抗函数的所有系数都是实数,因此其分子和分母多项式的根将以共轭复数的形式出现,且分子和分母多项式所有的根均位于单位圆上而且彼此轮流出现。这样,$I_p(z)$就一共有 $p-1$ 个极点和零点位于单位圆上(不包括零点和(或)极点为 1 和-1 的情况)。因为当 p 为奇数时,可以证明 $P_1(z)$有一个 $z=-1(\omega=\pi)$的根,$Q_1(z)$有一个 $z=1(\omega=0)$的根,则 $P_1(z)$和 $Q_1(z)$各有$(p-1)/2$ 个共轭复根。当 p 为偶数时,$Q_1(z)$有 $z=\pm1$ 两个实根,则 $P_1(z)$和 $Q_1(z)$各有 $p/2$ 以及 $p/2-1$ 个共轭复根。在 AMR-WB 中,$p=16$,其中,$P_1(z)$有 8 个共轭复根,$Q_1(z)$有 7 个共轭复根,且都位于在单位圆上,另外 $Q_1(z)$还有 $z=\pm1$ 两个实根。此处针对 p 为偶数进行讨论。这样 $P_1(z)$和 $Q_1(z)$可进一步表示为

$$P_1(z)=\prod_{i\in\{1,3,\cdots,p-1\}}(1-\mathrm{e}^{\mathrm{j}\omega_i}z^{-1})(1-\mathrm{e}^{-\mathrm{j}\omega_i}z^{-1})=\prod_{i\in\{1,3,\cdots,p-1\}}(1-2\cos\omega_i z^{-1}+z^{-2}) \tag{6.53}$$

$$\begin{aligned}Q_1(z)&=(1-z^{-2})\prod_{i\in\{2,4,\cdots,p-2\}}(1-\mathrm{e}^{\mathrm{j}\omega_i}z^{-1})(1-\mathrm{e}^{-\mathrm{j}\omega_i}z^{-1})\\&=(1-z^{-2})\prod_{i\in\{2,4,\cdots,p-2\}}(1-2\cos\omega_i z^{-1}+z^{-2})\end{aligned} \tag{6.54}$$

由此引入新的多项式

$$p'_1(z)=P_1(z) \tag{6.55}$$

$$q'_1(z)=\frac{Q_1(z)}{1-z^{-2}} \tag{6.56}$$

因为多项式 $p'_1(z)$和 $q'_1(z)$在单位圆上($\mathrm{e}^{\pm\mathrm{j}\omega_i}$)分别有 $p/2$ 和 $p/2-1$ 个共轭复根,因此可得到如下的多项式 $P'_1(z)$和 $Q'_1(z)$

$$\begin{aligned}P'_1(z)&=(1+k_p)\prod_{i\in\{1,3,\cdots,p-1\}}(1-\mathrm{e}^{\mathrm{j}\omega_i}z^{-1})(1-\mathrm{e}^{-\mathrm{j}\omega_i}z^{-1})\\&=(1+k_p)\prod_{i\in\{1,3,\cdots,p-1\}}(1-2\cos\omega_i z^{-1}+z^{-2})\end{aligned} \tag{6.57}$$

$$\begin{aligned}Q'_1(z)&=(1+k_p)\prod_{i\in\{2,4,\cdots,p-2\}}(1-\mathrm{e}^{\mathrm{j}\omega_i}z^{-1})(1-\mathrm{e}^{-\mathrm{j}\omega_i}z^{-1})\\&=(1+k_p)\prod_{i\in\{2,4,\cdots,p-2\}}(1-2\cos\omega_i z^{-1}+z^{-2})\end{aligned} \tag{6.58}$$

其中,$\cos\omega_i$,$i=1,2,\cdots,p-1$,是 ISP 前 $p-1$ 个系数在余弦域的表示,式(6.57)、式(6.58)中的 k_p 是 ISP 的最后一个系数,也称为常数增益,这样 ISP 的 p 个参数可以表示如下:$\cos\omega_1$,$\cos\omega_2$,$\cdots$,$\cos\omega_{p-1}$,k_p。其中的 ω_1,ω_2,$\cdots$,ω_{p-1}是与前 $p-1$ 个 ISP 系数相对应的频率,且按升序排列,即 $0<\omega_1<\omega_2\cdots<\omega_{p-1}<\pi$,常数增益 k_p 满足 $|k_p|<1$,这样可以使 LPC 滤波器的稳定性得到保证。在 AMR-WB 中,取 $k_p=a_p$。对 k_p 进行相应的反余弦变换可得 $\frac{1}{2}\arccos k_p$。这样就可得到与这 p 个 ISP 系数相对应的频率,称为导抗谱频率 ISF。

由 ISP 的定义可见,ISP 包含两种不同的参数,其中前 $p-1$ 个 ISP 由 p 阶对称和反对称多项式的共轭复根组成,最后一个系数为常数增益。前 $p-1$ 个 ISP 表现出与 p 个 LSP 相似的一些特性:

① 都在单位圆上且满足降序排列的特性,这主要是由于式(6.32)、式(6.33)与式(6.50)、式(6.51)基本相同。

② 与 ISP 对应的前 $p-1$ 个 ISF 都满足升序排列的顺序特性,且 ISP 的第 p 个系数 $|k_p|<1$,这使得与之对应的 LPC 滤波器的稳定性可以得到保证。因此 ISP 分析就是用 $p-1$ 个离散频率和离散频率的分布密度来表示语音信号频谱特性的方法。

③ 帧内 ISP 参数具有相对独立的性质,相邻帧 ISP 参数之间则具有较强的相关性,这有利于语音编码时帧间参数的量化和内插。

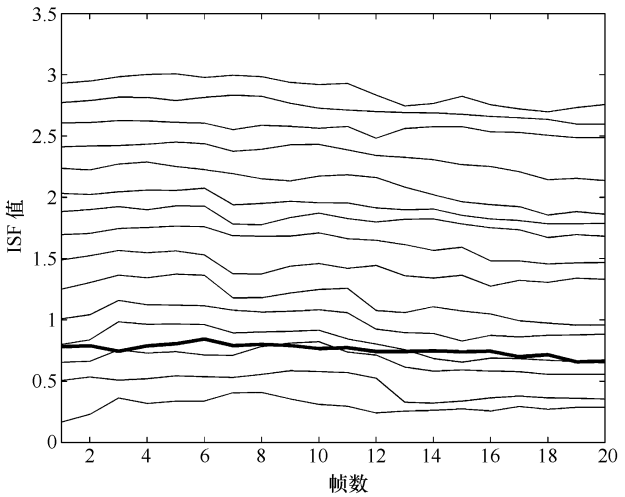
④ ISP 参数能够反映声道幅度谱的特点,在幅度大的地方分布较密,反之较疏。这样就相当于反映出了幅度谱中的共振峰特性。按照 LPC 分析的原理,语音信号的谱特性可以由 LPC 模型谱来估计,将式(6.50)、式(6.51)相加可得

$$A(z)=\frac{1}{2}[P_1(z)+Q_1(z)] \tag{6.59}$$

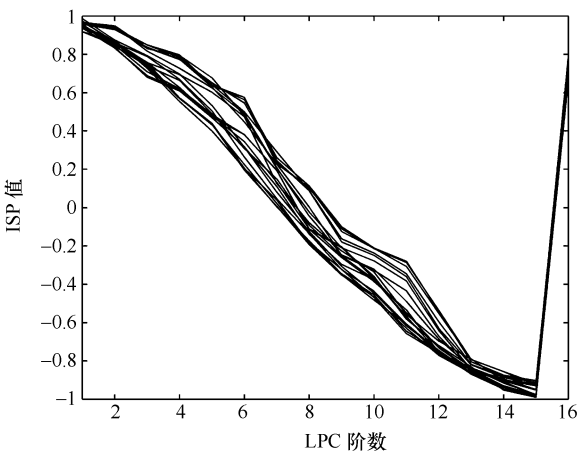
这样，功率谱可以表示为

$$|H(e^{j\omega})|^2 = \frac{1}{|A(e^{j\omega})|^2} = 4|P_1(e^{j\omega}) + Q_1(e^{j\omega})|^{-2}$$
$$= 2^{-p+2}\left[\prod_{i\in\{1,3,\cdots,p-1\}}(\cos\omega - \cos\omega_i)^2 + \sin^2\omega \prod_{i\in\{2,4,\cdots,p-2\}}(\cos\omega - \cos\omega_i)^2\right]^{-1} \tag{6.60}$$

图 6.6 给出了 $p=16$ 时连续 20 帧语音的 ISP 参数及 ISF 轨迹图。其中图 6.6(a)为连续 20 帧语音信号的 ISF 轨迹图，图中 15 条细实线为连续 20 帧语音前 15 个 ISF 参数形成的 ISF 变化轨迹图，按从下往上的顺序分别对应前 $p-1$ 个 ISF 参数 ω_i，$i=1,2,\cdots,p-1$。第 16 个 ISF 参数对应图中那条黑粗线，可见该参数已经不满足顺序特性。图 6.6(b)为连续 20 帧语音信号的 ISP 图。可见前 $p-1$ 个 ISP 参数满足降序排列特性，但最后一个 ISP 参数也不满足顺序特性。



(a) 连续 20 帧语音信号的 ISF 轨迹图



(b) 连续 20 帧语音信号的 ISP 波形

图 6.6　连续 20 帧语音信号的 ISP 图及 ISF 轨迹图

6.7.2 LPC 与 ISP 参数间的转换及 MATLAB 实现

在进行语音编码时，也可将 LPC 系数转换为 ISP 系数以进行量化和内插。由于 ISP 参数与 LSP 参数仅第 p 个参数不同，因此 LPC 系数与 ISP 系数之间的转换与 LSP 类似。从 LPC 转换为 ISP 系数时，首先应用求解 LSP 参数的方法求解出前 $p-1$ 个 ISP 系数，再给第 p 个参数赋上合适的值，即可得到 ISP 系数。解码时，首先根据量化的 ISP 系数得到 $p-1$ 个 LPC 系数，其求解方法同 6.6.3 节，再根据第 p 个 ISP 系数得到最后一个 LPC 系数即可。下面分别给出从 LPC 转换为 ISP 系数的 MATLAB 程序以及将 ISP 系数转换为 LPC 的 MATLAB 程序。

【程序 6.4】a_isf_lpc_conversion.m

```
% 已知语音文件求出其 LPC 系数后，求其对应的 ISF，再将 ISF 转换为 a。其中 isf 为转换后的
% ISF 值，a 为 isf 转换后的 lpc 系数
clear;close all
clc
fid=fopen('sx86.txt','r');
p1=fscanf(fid,'% f')
fclose(fid);
p=filter([1 -0.68], 1, p1);% 预加重滤波
x=fra(320,160,p);
x=x(60,:);
a3=lpc(x,15);
a4=a3(:); % 将线性预测系数赋给矩阵 a
lsf=a_lsf_conversion(a4) ; % 调用函数 a_lsf_conversion 实现从 LPC 系数到 lsf 参数的
                          % 转换
% 函数 a_lsf_conversion()的 MATLAB 程序见本章 6.6.2 节所赋程序
% 此处也可调用 MATLAB 自带的库函数 lsf=poly2lsf(a4) ;
isf=lsf;
isf(16,1)=0.5*acos(a4(16,1));
% isp 的最后一个参数取为 a 的最后一个参数，isf 的最后一个参数取为 0.5*acos(a4(16,1))

% 下面是从 isf 求 a 的程序，其中前 p-1 个 a 参数根据前 p-1 个 isf 参数得到，最后一个 a 参
% 数根据 isf 的第 p 个参数得到
isf1=isf(1:(size(isf)-1),:);% 将 isf 前 p-1 个参数赋给 isf1
a2=lsf_lpc_conversion(isf1) % 调用函数 lsf_lpc_conversion 实现从 lsf 参数到 LPC 系
                            % 数的转换
a2(1,16)=cos(2*isf(16,1));% 最后一个 a 参数根据 isf 的第 p 个参数得到
a=a2;% 将转换得到的 lpc 系数赋给 a
% EOF a_isf_lpc_conversion.m
```

绘制图 6.6 的 MATLAB 程序见程序 6.5：a_isf_lpc_conversion_20frame.m

【程序 6.5】a_isf_lpc_conversion_20frame.m

```
% 求连续 20 帧语音的 16 阶 ISF 轨迹图
% a_isf_lpc_conversion_20frame.m
clear;close all
```

```
clc
fid=fopen('sx86.txt','r');
p1=fscanf(fid,'% f')
fclose(fid);
p=filter([1 -0.68], 1, p1);% 预加重滤波
x1=fra(320,160,p);
for i=60:79
x=x1(i,:)
a3=lpc(x,15);
a4=a3(:);% 将线性预测系数赋给矩阵 a4
lsf=a_lsf_conversion(a4) % 调用函数 a_lsf_conversion 实现从 LPC 系数到 lsf 参数的
                         % 转换
isf=lsf;
isf(16)=0.5*acos(a4(16,1));% isp 的最后一个参数取为 a 的最后一个参数,isf 的最后一
                           % 个参数取为 0.5*acos(a4(16,1))
isp=cos(isf);
hold on% 让连续 20 帧 isp 及 isf 绘制在一个图形中
figure(1);
    for j=1:16
    isf2(i-59,j)=isf(j);
    end
figure(2);
plot(isp)
end
% EOF a_isf_lpc_conversion_20frame.m
```

运行 a_isf_lpc_conversion_20frame.m 程序后，在 figure(2)可得到连续 20 帧语音的 ISP 图。

绘制连续 20 帧 isf 变化轨迹方法为：在 Workspace 中选中 isf2，单击图标 plot 按钮，并以 figure(1)作为绘图区域即可。

6.8 LPC 导出的其他语音参数

在线性预测语音编码过程中，如果直接在信道传输线性预测滤波器系数，则对误差会非常敏感，导致一个小的误差使整个频谱质量下降，甚至使线性预测滤波器变得不稳定。因此在语音编码算法中，通常将线性预测滤波器系数转换为与之等效的参数，再进行量化编码。这些参数一般是由线性预测滤波器系数推演出来的，因而称之为线性预测的推演参数。这些推演参数除了 LSP、ISP 之外，还包括反射系数、对数面积比系数、LPC 倒谱等，它们各有不同的物理意义和特性，例如量化特性、插值特性和参数灵敏度等。下面分别进行介绍。

6.8.1 反射系数

反射系数也称为部分相关系数，即 PARCOR 系数，用 k_i 表示。由于它是与多节级联无损声管模型中的反射波相联系的，因而通常称之为反射系数。已知线性预测系数 a_i，$i=1,2,\cdots,$

p，求反射系数 k_i 的递推过程如下：

$$\begin{cases} a_j^{(p)} = a_j & 1 \leqslant j \leqslant p \\ k_i = a_i^{(i)} & \\ a_j^{(i-1)} = [a_j^{(i)} + a_i^{(i)} a_{i-j}^{(i)}]/(1-k_i^2) & 1 \leqslant j \leqslant i-1 \end{cases} \tag{6.61}$$

反过来，已知反射系数 k_i，求相应的线性预测系数 a_i，$i=1,2,\cdots,p$ 的递推过程如下：

$$\begin{gathered} a_i^{(i)} = k_i \\ a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \quad 1 \leqslant j \leqslant i-1 \\ a_j = a_j^{(p)} \quad 1 \leqslant j \leqslant p \end{gathered} \tag{6.62}$$

为了保证相应的线性预测合成滤波器的稳定性，反射系数 k_i 通常取为 $-1 \leqslant k_i \leqslant 1$。但是 k_i 具有不平坦的频谱灵敏度，其靠近 1 的值比远离 1 的值需要更高的量化精度。因此需要将 k_i 进行非线性变换，下面的对数面积比系数就是广泛采用的一种非线性函数。

6.8.2 对数面积比系数 LAR

由反射系数 k_i 可进一步推导出对数面积比系数，其定义为

$$g_i = \log(A_{i+1}/A_i) = \log[(1-k_i)/(1+k_i)] \quad 1 \leqslant i \leqslant p \tag{6.63}$$

对上式两边取以 e 为底的指数，整理可得

$$k_i = (1-\exp(g_i))/(1+\exp(g_i)) \quad 1 \leqslant i \leqslant p \tag{6.64}$$

其中，A_i 是多节级联无损声管模型中第 i 节的截面积。由于 g_i 相对于谱的变化的灵敏度比较平缓，因而特别适合量化。但是采用 LAR 量化时，要想使频谱失真最小，每一个系数大约需要 4 个 bit 进行编码，这将占编码器容量的一大部分。另外用 LAR 表示时，LPC 参数帧与帧之间的相关性将不再显著。鉴于此以及 LSF 参数所具有的帧到帧的优良的内插特性，在语音编码系统中 LAR 渐渐被 LSF 参数取代。

6.8.3 LPC 倒谱及其 MATLAB 实现

线性预测倒谱系数 LPCC 是 LPC 系数在倒谱域中的表示。语音信号的倒谱指的是这个信号 z 变换的对数模函数的反 z 变换。这样，通过对语音信号的傅里叶变换取模的对数再求反傅里叶变换即可得到一个信号的倒谱。信号的倒谱也是描述语音信号特性的一个较好的参数，其特征基于语音信号为自回归信号的假设。LPCC 参数的优点是计算量小，易于实现，对元音有较好的描述能力，缺点是对辅音的描述能力较差，抗噪性能较差。由于线性预测合成滤波器的频率响应 $H(e^{j\omega})$ 可以反映声道的频率响应及被分析信号的谱包络，因此可以用 $\log|H(e^{j\omega})|$ 做反傅里叶变换求出倒谱系数。设通过线性预测分析得到的声道模型系统函数为

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{6.65}$$

其冲激响应为 $h(n)$，倒谱为 $\hat{h}(n)$，则有

$$\hat{H}(z) = \ln H(z) = \sum_{n=1}^{\infty} \hat{h}(n) z^{-n} \tag{6.66}$$

将式(6.65)代入式(6.66)并将其两边对 z^{-1} 求导,整理可得

$$\left(1-\sum_{i=1}^{p}a_i z^{-i}\right)\sum_{n=1}^{\infty}n\,\hat{h}(n)z^{-n+1}=\sum_{i=1}^{p}ia_i z^{-i} \tag{6.67}$$

令上式两边的各次 z^{-1} 的系数分别相等,可得由 LPC 系数求倒谱系数的递推公式:

$$\hat{h}(n)=\begin{cases}a_n & n=1\\ a_n+\sum_{k=1}^{n-1}k\,\hat{h}(k)a_{n-k}/n & 1<n\leqslant p+1\\ \sum_{k=1}^{n-1}k\,\hat{h}(k)a_{n-k}/n & n>p+1\end{cases} \tag{6.68}$$

由于线性预测合成滤波器的极点在单位圆内,其所对应的单位冲击响应是一个最小相位序列。因此其倒谱系数是一个右半序列。

由于语音信号的倒谱能较好地描述语音的共振峰特征,并比较彻底地去掉了语音产生过程中的激励信息,因此在语音识别系统中得到了较好的应用效果。实验表明,使用倒谱可以提高特征参数的稳定性。下面给出从 LPC 系数求 LPCC 参数的 MATLAB 程序。

【程序 6.6】 a_lpcc_main.m

```
% a_lpcc_main()
% 已知语音文件求出其 LPC 系数后,调用 lpc_lpcc_conversion()函数求 lpcc 参数
% 结果输出到文件"从 lpc 系数解得的 LPCC 参数.txt"
clear;close all
clc
fid=fopen('sx86.txt','r');
p1=fscanf(fid,'% f')
fclose(fid);
p=filter([1 -0.68], 1, p1)% 预加重滤波
x=fra(320,160,p)% 将 p 进行分帧,帧长 320,帧重叠 160
x=x(60,:)
a1=lpc(x,16)
a=a1(:);% 将线性预测系数赋给矩阵 a
a_num=16;% a_num 为线性预测系数阶次,不包括 a(0)=1
C_num=16;% C_num 为线性预测倒谱系数 LPCC 个数
lpcc=lpc_lpcc_conversion(a,C_num,a_num)% 调用 lpc_lpcc_conversion()函数求 lpcc 参数
% 结果输出到文件"从 lpc 系数解得的 LPCC 参数.txt"
fid= fopen('从 lpc 系数解得的 LPCC 参数.txt','w');
fprintf(fid,'lpc 系数:\n');
fprintf(fid,'% 6.2f, ',a);
fprintf(fid,'\n');
fprintf(fid,'从 lpc 系数解得的 LPCC 参数:\n');
fprintf(fid,'% 8.4f, ',lpcc);
fclose(fid);
% EOF a_lpcc_main()
```

求 LPCC 参数的函数见程序 lpc_lpcc_conversion.m。

```
% 计算倒谱参数 C(1)到 C(C_num)的函数
```

```
% 其中 a 为 lpc 系数,a_num 为 LPC 系数个数,即 LPC 系数阶次,不包括 a(0)=1;
% C_num 为倒谱系数个数
%  lpc_lpcc_conversion.m
function lpcc=lpc_lpcc_conversion(a,C_num,a_num)
n_lpc=a_num;n_lpcc=C_num;
lpcc=zeros(n_lpcc,1); % 初始化 lpcc 矩阵为 n_lpcc 行 1 列的一个全 0 矩阵
lpcc(1)=a(1); % C(1)=a(1)
% 计算倒谱参数 C(2)到 C(n_lpc)
for n=2:n_lpc
    lpcc(n)=a(n);
    for m=1:n-1
        lpcc(n)=lpcc(n)+a(m) * lpcc(n-m) * (n-m)/n;
    end
end
% 计算倒谱参数 C(n_lpc+1)到 C(C_num)
for n=n_lpc+1:n_lpcc
    lpcc(n)=0;
    for m=1:n_lpc
        lpcc(n)=a(n)+lpc(m) * lpcc(n-m) * (n-m)/n;
    end
end
% EOF lpc_lpcc_conversion.m
```

图 6.7 给出了语音信号及其 LPC 谱包络与倒谱包络的比较，由图可以看出，虽然共振峰频率在两个图中都明显可见，且两者得到的谱峰个数相同，但一般情况下语音信号 LPC 谱比倒谱包络的共振峰要少，这是由于 LPC 分析的阶数决定其共振峰的个数，但倒谱不存在这种限制。

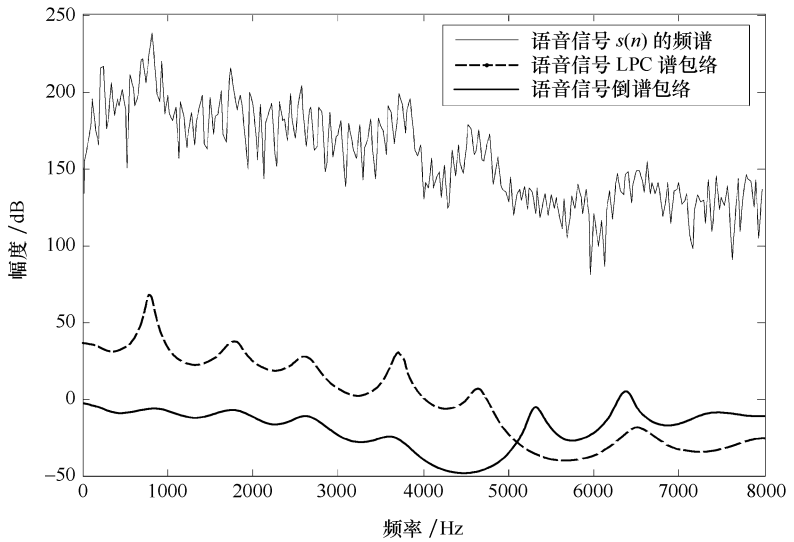


图 6.7　语音信号及其 LPC 谱包络与倒谱包络的比较

6.9 LPC 分析的频域解释

由于语音产生模型中的全极点滤波器是声门、声道和嘴唇辐射的综合模拟，所以其频率特性主要反映了声道的共振特性。而语音信号的 LPC 系数就是语音信号产生模型中全极点合成滤波器 $H(z)$的分母多项式的系数，因此当根据一帧语音的取样值计算出语音信号的 LPC 系数后，只要将 $z=e^{j\omega}$代入 $H(z)$进行计算，就意味着求得了这帧语音信号产生模型的频率特性，即有

$$H(e^{j\omega}) = \frac{G}{1-\sum_{i=1}^{p} a_i e^{-j\omega i}} = \frac{G}{A(e^{j\omega})} \tag{6.69}$$

如果我们画出 $H(e^{j\omega})$随频率变化的波形，可以预料在共振峰频率上会出现峰起，因此 LPC 分析可以看成是对语音信号的短时谱进行估计的一种有效方法。在语音产生模型中，语音的功率谱等于激励源功率谱与全极点合成滤波器频率特性模的平方的乘积，而激励源是准周期冲击序列或白噪声，其功率谱是平坦的。所以语音的功率谱主要由全极点滤波器的特性来决定。

6.9.1 最小预测误差的频域解释

由均方预测误差 $E[e^2(n)]$及 Parseval 定理知均方预测误差的频域表示式，即功率谱为

$$\begin{aligned} E_p &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |E(e^{j\omega})|^2 d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |S(e^{j\omega})|^2 |A(e^{j\omega})|^2 d\omega = \frac{G^2}{2\pi}\int_{-\pi}^{\pi} \frac{|S(e^{j\omega})|^2}{|H(e^{j\omega})|^2} d\omega \end{aligned} \tag{6.70}$$

其中，$S(e^{j\omega})$是语音 $s(n)$的傅里叶变换，而 $A(e^{j\omega}) = 1-\sum_{i=1}^{p} a_i e^{-j\omega i}$。式(6.70)表明，使 E_p 为最小，等效于使语音的能量谱对 $|H(e^{j\omega})|^2$ 比值的积分为最小。这样，LPC 分析在频域上可理解为：给定语音信号的谱 $|S(e^{j\omega})|^2$，期望用一个 p 阶全极点滤波器作为其模型，该模型输出的谱 $|H(e^{j\omega})|^2$ 使比值 $|S(e^{j\omega})|^2/|H(e^{j\omega})|^2$ 的积分最小。

如果全极点滤波器冲激响应的能量等于语音信号的能量，就意味着全极点滤波器冲激响应自相关函数的前$(p+1)$个系数等于语音信号自相关函数的前$(p+1)$个系数。这样，当 $p\to\infty$时，这两个自相关函数在所有值上皆相等，因此

$$\lim_{p\to\infty} |H(e^{j\omega})|^2 = |S(e^{j\omega})|^2 \tag{6.71}$$

式(6.71)表明，如果 p 足够大，则我们就能以任意小的误差用全极点模型来逼近信号谱。

注意，即使 $p\to\infty$时，上式表明 $|H(e^{j\omega})|^2=|S(e^{j\omega})|^2$，但是式 $H(e^{j\omega})=S(e^{j\omega})$并不一定成立，即模型谱的频率响应不一定等于信号的傅里叶变换，因为 $S(e^{j\omega})$不一定是最小相位的，而 $H(e^{j\omega})$必定是最小相位的，这是因为 $H(e^{j\omega})$是一个全极点滤波器的转移函数，其极点应全部位于单位圆内。

6.9.2 LPC 谱估计

为了表明用线性预测谱作为语音信号谱的能力，下面对 $20\lg|H(e^{j\omega})|$和 $20\lg|X(e^{j\omega})|$进行比较。其中信号谱由 FFT 分析得到，$H(e^{j\omega})$是用自相关法得到的 16 个极点的 LPC 谱。由

图 6.8 可以看出，在信号能量较大的区域（即在信号谱的峰值处），LPC 谱和信号谱匹配得很好，而在信号能量较低的区域（（即在信号谱的谷底处），则匹配得较差。这可由式(6.70)进行说明，式(6.70)表明，按最小均方误差求解时，$|S(e^{j\omega})|>|H(e^{j\omega})|$的区域在总误差中所起的作用比$|S(e^{j\omega})|<|H(e^{j\omega})|$的区域大。因此 LPC 谱误差准则有利于在谱峰附近的良好匹配，而在谱谷附近匹配得较差。

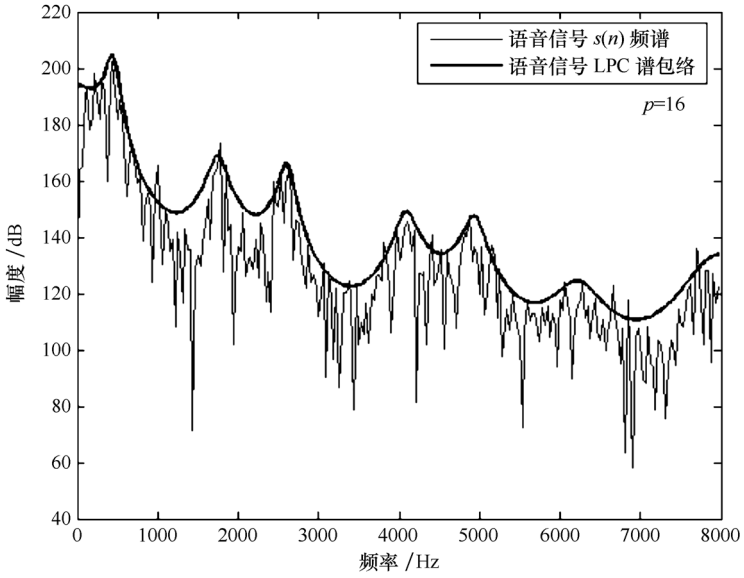
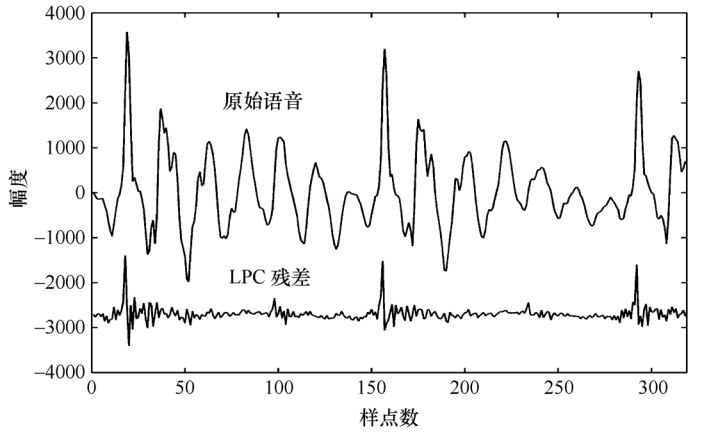


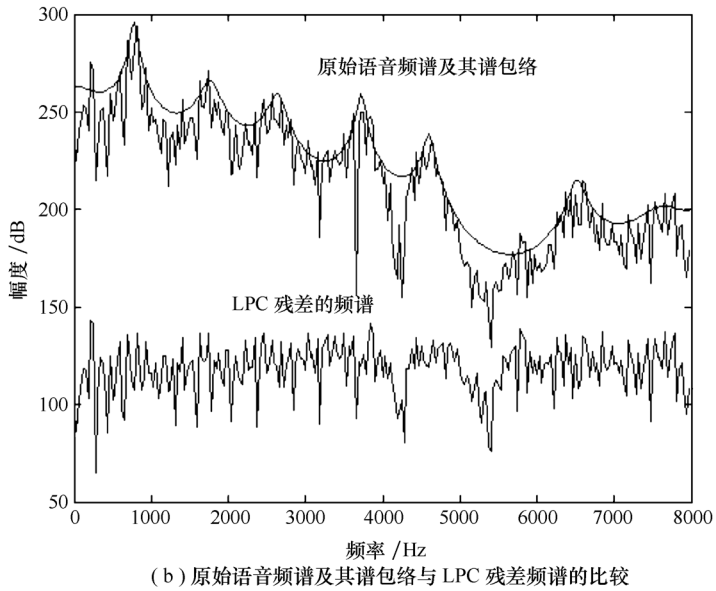
图 6.8　LPC 谱与实际谱的比较

图 6.9 给出了原始语音信号和经过 LPC 逆滤波语音的时域和频域波形图。其中图 6.9 (a)为时域波形，图 6.9(b)为语音频谱及其谱包络和 LPC 残差频谱波形。可见，与原始信号相比，经过 LPC 逆滤波后所得到的误差信号有比较小的变化，且误差信号的频谱是很平坦的。

由上面的分析可知，线性预测分析的阶数 p 可有效控制所得谱的平滑度。这可由图 6.10 来说明，图中给出了语音信号的功率谱以及各个不同阶数的线性预测谱。显然，p 越大，有更多的谱细节可被保存下来，但我们的目的只是要得到声门激励、声道以及辐射联合效应谱的表示式，这样，LPC 模型阶数 p 的选择，就应该从频谱估计精度、计算量、存储量等多方面综合进行考虑。其阶次 p 的选择，首先要保证有足够的极点来模拟声道响应的谐振结构，因此主要取决于采样频率而基本上和所用的 LPC 分析方法无关。根据对发声过程机理的分析，被分析的语音谱一般可以用每千赫兹具有两个极点（可以是一对复共轭极点）的平均密度来表示声道造成的响应，因此当采样频率是 F_s（kHz）时，为了表示语音谱，总共需要 F_s 个极点。例如当采样频率为 16kHz 时，为了表示声道响应需要 16 个极点，此外需要 3～4 个极点来恰当的表示激励源的谱和辐射的组合效应，因此在 16kHz 采样情况下，要求 p 值约为 18～20。为了说明这个结论，图 6.11 给出了在 16kHz 采样率时，清音和浊音语音相对应的归一化预测误差随阶数 p 的变化曲线。可见，虽然阶数 p 增加时预测误差总是下降的，但当 p 达到 18 以后，误差变化基本趋于平缓，这说明 p 值再进一步增加时，误差减小已经不明显。且由图 6.11 还可看出，清音的归一化预测误差比浊音高得多，由此可进一步说明，全极点模型对清音来说远没有浊音那样精确。



(a) 原始语音和 LPC 逆滤波语音的波形图



(b) 原始语音频谱及其谱包络与 LPC 残差频谱的比较

图 6.9　原始语音及其 LPC 残差在时域和频域的波形比较

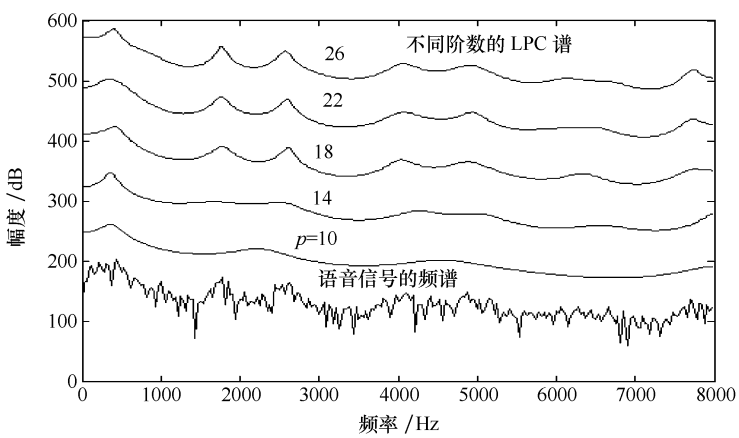


图 6.10　以 16kHz 采样的语音信号,预测器阶数不同时所得到的谱

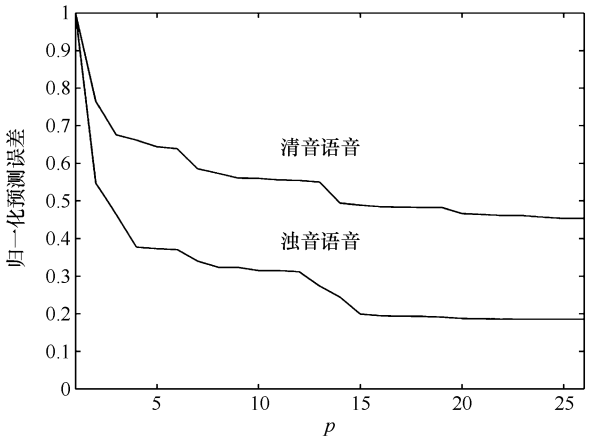


图 6.11　归一化预测误差与预测器阶数的变化关系

第7章　矢量量化

7.1　概　　述

随着计算机及数字通信技术的高速发展，人类之间交流的信息日益丰富，包括语音、文本、图像、视频等。这些信息变换成信号后，必须通过一定的系统进行传输或加工处理。数字通信系统以其抗干扰能力强，保密性好，便于传输、存储、交换和处理等优点得到广泛应用，但数字信号的数据量通常很大，给存储器的存储容量、通信信道的带宽及计算机的处理速度带来压力，因此必须对其量化压缩。量化可以分为两大类：一类是标量量化，另一类是矢量量化 VQ。标量量化是把抽样后的信号值逐个进行量化，而矢量量化是先将 $k(k\geqslant 2)$ 个抽样值形成 k 维空间 R^k 中的一个矢量，然后将此矢量进行量化，并设法使其失真或量化噪声最小，它可以极大地降低数码率，优于标量量化。各种数据都可以用矢量表示，直接对矢量进行量化，可以方便地对数据进行压缩。矢量量化属于不可逆压缩方法，能够有效地利用矢量中各分量间相互关联的性质(线性依赖性、非线性依赖性、概率密度函数的形状及矢量维数)以消除冗余度，具备比特率低、解码简单、失真较小的优点。矢量量化压缩技术不但广泛应用于图像和语音压缩编码等传统领域，而且在移动通信、语音识别、文献检索及数据库检索等领域得到越来越广泛的应用。

矢量量化的理论基础是香农的率－失真理论。率－失真理论是对给定的失真 D，可以计算率－失真函数 $R(D)$，$R(D)$ 定义为：在给定的失真 D 条件下，所能够达到的最小速率(用每维计算)；或者反过来，可以计算率－失真函数的逆函数 $D(R)$，称 $D(R)$ 为失真－率函数，它定义为：在给定的速率(以 bit/s 计算)条件下所能够达到的最小失真。$D(R)$ 或 $R(D)$ 所给出的编码速率极限，不仅适用于矢量量化，而且适用于所有信源编码方法。$D(R)$ 是在维数 $k\to\infty$ 时 $D_k(R)$ 的极限，即

$$D(R)=\lim_{k\to\infty}D_k(R)$$

率－失真理论指出，利用矢量量化，编码性能有可能任意接近率－失真函数，其方法是增加维数 k。率－失真理论在实际应用中有重要指导意义：率－失真函数常常作为一个理论下界与实际编码速率相比较，分析系统还有多大的改进余地。如果某系统的最高性能都不能满足系统或客户的要求，人们就不必浪费精力用给定的参数来设计出一个实际系统，因为永远设计不出满足要求的系统，除非降低系统的某项性能指标。相反，如果一个实际系统的性能已经接近于理论上界，则不应再投入更多的资金和时间来追求微不足道的改善。如果某系统的性能优于理论上界，则必须怀疑该系统模型的准确性。总之，率－失真理论指出了矢量量化的优越性。但是，率－失真理论是一个存在性定理而非构造性定理，因为它没有指出如何构造矢量量化器。

1956 年 Steinhaus 第一次系统地阐述了最佳矢量量化问题。1957 年在 Loyd 的“PCM 中的最小平方量化”一文中给出了如何划分量化区间和如何求量化值问题的结论。与此同时，Max 也得出了同样的结果。虽然他们谈论的都是标量量化问题，但他们的算法对后来的矢量

量化的发展有着深刻的影响。1978 年，Buzo 第一个提出实际的矢量量化器。他提出的量化系统的组成分为两步：第一步将语音做线性预测分析，求出预测系数；第二步对这些系数做矢量量化，于是得到压缩数码的语音编码器。1980 年，Linde、Buzo 和 Gray 将 Lloyd-Max 算法推广，发表了第一个矢量量化器的设计算法，通常称为 LBG 算法。这就使矢量量化的研究向前推进了一大步。这一时期，人们对矢量量化的问题展开了全面的研究，其中主要的是对失真测度的探讨，码书的设计，各种矢量量化系统的研究，快速搜索算法的寻找等。

矢量量化的效果是很明显的，1980 年美国加州公司的 Wong 和 Juang 等人在原来编码速率为 2.4kbit/s 的线性预测声码器上，仅将滤波系数由标量量化改为矢量量化，就可使编码速率降低到 800bit/s，而声音质量基本未下降。1983 年美国 BBN 公司的 Makhoul 等人研制了一种分段式声码器。由于该声码器采用了矢量量化，所以可以用 150bit/s 的速率来传送可懂的语音。近几十年来在已经提出的各种矢量量化方法和系统的基础上，再与其他编码技术相结合，得到了更好的矢量量化方法，用硬件实现矢量量化系统的方法也日益增多。

7.2 矢量量化基本原理

7.2.1 矢量量化的定义

矢量量化是先把信号序列的每 K 个连续样点分成一组，形成 K 维欧氏空间中的一个矢量，然后对此矢量进行量化。

如图 7.1 中的输入信号序列$\{x_n\}$，每 4 个样点构成一个矢量(取 $K=4$)，共得到 $n/4$ 个四维矢量，$\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3,\cdots,\boldsymbol{X}_{n/4}$。矢量量化就是先集体量化 $\boldsymbol{X}_1$，然后再量化 $\boldsymbol{X}_2$，依次向下量化，下面以 $K=2$ 为例进行说明。

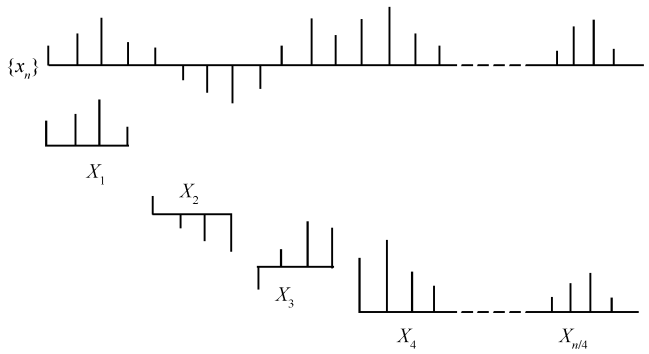


图 7.1 四维矢量形成示意图

当 $K=2$ 时，所得到的是一些二维矢量。所有可能的二维矢量就形成了一个平面。如果记二维矢量为(a_1,a_2)，所有可能的(a_1,a_2)就是一个二维欧氏空间。如图 7.2(a)所示，矢量量化就是先把这个平面划分成 N 块(相当于标量量化中的量化区间) S_1，$S_2,\cdots,S_N$，然后从每一块中找一个代表值 $\boldsymbol{Y}_i(i=1,2,\cdots,N)$(相当于标量量化中的量化值)，这就构成了一个有 N 个区间的二维矢量量化器。图 7.2(b)所示的是一个 7 区间的二维矢量量化器，即 $K=2,N=7$，共有 $\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_7$ 7 个代表值，通常把这些代表值 $\boldsymbol{Y}_i$ 称为量化矢量。

若要对落在二维矢量空间里的一个模拟矢量 $\boldsymbol{X}=(a_1,a_2)$进行量化，首先要选一个合适的失真测度，而后利用最小失真原则，分别计算用量化矢量 $\boldsymbol{Y}_i(i=1,2,\cdots,7)$替代 $\boldsymbol{X}$ 所带来的失

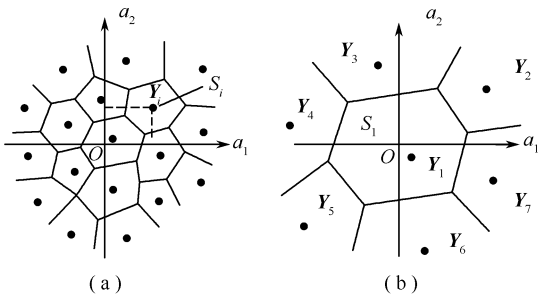


图 7.2　矢量量化示意图

真。其中最小失真值所对应的那个量化矢量 $\boldsymbol{Y}_i(i=1,2,\cdots,7)$ 中某一个，就是模拟矢量 $\boldsymbol{X}$ 的重构矢量（或称恢复矢量）。通常把所有 N 个量化矢量（重构矢量或恢复矢量）构成的集合 $\{Y_i\}$ 称之为码书（codebook）或码本。码书中的矢量称之为码字（codeword）或码矢（codevector）。例如图 7.2(b) 中所示的矢量量化器的码书 $\mathscr{Y}=\{\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_7\}$，其中每个量化矢量 $\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_7$ 称为码字或码矢。不同的划分或不同的量化矢量选取就可以构成不同的矢量量化器。

根据上面对矢量量化的描述，我们可以把矢量量化定义为

$$\boldsymbol{Y}\in\mathscr{Y}_N=\{\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_N\,|\,\boldsymbol{Y}_i\in R^k\}$$

矢量量化是把一个 K 维模拟矢量 $\boldsymbol{X}\in\mathscr{X}\subset R^k$ 映射为另一个 k 维量化矢量

其数学表达式为

$$\boldsymbol{Y}=Q(\boldsymbol{X}) \tag{7.1}$$

式中　$\boldsymbol{X}$——输入矢量

$\mathscr{X}$——信源空间

R^k——k 维欧氏空间

$\boldsymbol{Y}$——量化矢量（码字或码矢）

$\mathscr{Y}_N$——输出空间（即码书）

$Q(\cdot)$——量化符号

N——码书的大小（即码字的数目）

矢量量化系统通常可以分解为两个映射的乘积

$$Q=\alpha\beta \tag{7.2}$$

式中，α 是编码器，它是将输入矢量 $\boldsymbol{X}\in\mathscr{X}\subset R^k$ 映射为信道符号集 $I_N=\{i_1,i_2,\cdots,i_N\}$ 中的一个元 i_j；β 是译码器，它是将信道符号 i_j 映射为码书中的一个码字 $\boldsymbol{Y}_i$。即

$$\alpha(\boldsymbol{X})=i_j \quad \boldsymbol{X}\in\mathscr{X},i_j\in I_N \tag{7.3}$$

$$\beta(i_j)=\boldsymbol{Y}_i \quad i_j\in I_N,\boldsymbol{Y}_i\in\mathscr{Y}_N \tag{7.4}$$

7.2.2　失真测度

设计矢量量化器的关键是编码器 $\alpha(\boldsymbol{X})$ 的设计，而译码器 $\beta(i)$ 的工作过程仅是一个简单的查表过程。设计编码器需引入失真测度的概念，失真测度的选择直接影响矢量量化系统的性能。

失真测度是以什么方法来反映用码字 $\boldsymbol{Y}_i$ 代替信源矢量 $\boldsymbol{X}$ 时所付出的代价。这种代价的统计平均值（平均失真）描述了矢量量化器的工作特性，即

$$D=E[d(X,Q(X))] \tag{7.5}$$

式中，$E[\cdot]$ 表示求期望。

常用的失真测度有如下几种。

① 平方失真测度

$$d(\boldsymbol{X},\boldsymbol{Y})=\|X-Y\|^2=\sum(\boldsymbol{X}_i-\boldsymbol{Y}_i)^2 \tag{7.6}$$

这是最常用的失真测度，因为它易于处理和计算，并且在主观评价上有意义，即小的失真值对应好的主观评价质量。

② 绝对误差失真测度

$$d(\boldsymbol{X},\boldsymbol{Y}) = |\boldsymbol{X}-\boldsymbol{Y}| = \sum_{i=1}^{k} |\boldsymbol{X}_i - \boldsymbol{Y}_i| \tag{7.7}$$

此失真测度的主要优点是计算简单，硬件容易实现。

③ 加权平方失真测度

$$d(\boldsymbol{X},\boldsymbol{Y}) = (\boldsymbol{X}-\boldsymbol{Y})^{\mathrm{T}}\boldsymbol{W}(X-Y) \tag{7.8}$$

式中　T——矩阵转置符号

$\boldsymbol{W}$——正定加权矩阵

在矢量量化器的设计中，失真测度的选择是很重要的。一般来说，要使所选用的失真测度有实际意义，必须要求它具有以下几个特点：

① 必须在主观评价上有意义，即小的失真对应好的主观质量评价。

② 必须在数学上易于处理，能导致实际的系统设计。

③ 必须可计算并保证平均失真 $D=E[d(X,Q(X))]$存在。

④ 采用的失真测度，应使系统容易用硬件实现。

7.2.3　矢量量化器

有了失真测度，就可以根据矢量量化的定义来具体设计矢量量化器了。通常用最小失真的方法——最近邻法 NNR 来设计，也就是要满足下式

$$\alpha(\boldsymbol{X}) = i \Leftrightarrow d(\boldsymbol{X},\boldsymbol{Y}_i) \leqslant d(\boldsymbol{X},\boldsymbol{Y}_j) \quad \forall j \in I_N \tag{7.9}$$

式中　$I_N=\{1,2,\cdots,i,\cdots,N\}$

N——码书的大小

符号⇔表示当且仅当(充分必要条件)。

这样就可以得到一个如图 7.3 所示的矢量量化器实现框图。其简单工作过程是：在编码端，输入矢量 $\boldsymbol{X}$ 与码书(Ⅰ)中的每一个或部分码字进行比较，分别计算出它们的失真。搜索到失真最小的码字 $\boldsymbol{Y}_i$ 的序号 i(或此码字所在码书中的地址)并将 i 的编码信号通过信道传输到译码端；在译码端，先把信道传来的编码信号译成序号 i，再根据序号 i(或码字 $\boldsymbol{Y}_i$ 所在地址)，从码书(Ⅱ)中查出相应的码字 $\boldsymbol{Y}_i$。由于码书(Ⅰ)与码书(Ⅱ)是完全一样的，此时失真 $D(\boldsymbol{X},\boldsymbol{Y}_i)$最小，所以 $\boldsymbol{Y}_i$ 就是输入矢量 $\boldsymbol{X}$ 的重构矢量(恢复矢量)。很明显，由于在信道中传输的并不是矢量 $\boldsymbol{Y}_i$ 本身，而是其序号 i 的编码信号，所以传输速率还可以进一步降低。

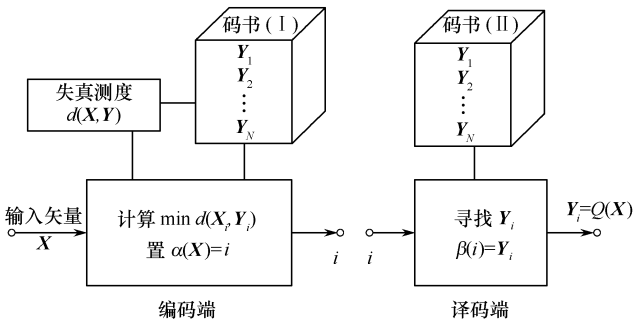


图 7.3　矢量量化原理框图

矢量量化是一种高效的数据压缩技术，和其他数据压缩技术一样，它除了有失真以外，还有一个传输速率问题，即每个样值（每维）平均编码所需的比特数。

矢量量化器的速率定义为

$$r=\frac{B}{K}=\frac{1}{K}\log_2 N \quad (\text{bit/样点或每维}) \tag{7.10}$$

式中 $B=\log_2 N$ 表示每个码字的编码比特数

N——码书的大小（即码字的数目）

K——维数

由式(7.10)可见，矢量量化器的速率 r 与码书大小 N 的对数 $\log_2 N$ 成正比，与维数 K 成反比。这说明 N 越大速率越高；而维数 K 越大，速率越低。

信道中传输速率 R_T 与矢量量化器速率 r 的关系为

$$R_T=f_s r \tag{7.11}$$

式中，f_s 为抽样速率。

7.3 最佳矢量量化器

在标量量化中，Lloyd-Max 算法给出了设计最佳标量量化器（失真最小）的两个必要条件：一是在预先划分好量化区间 $\Delta \boldsymbol{x}_\alpha(\alpha=1,2,\cdots,n)$ 情况下，集 $\{\hat{x}_\alpha\}$ 中每个量化值必须是相应量化区间的质量中心；二是当量化值 $\hat{x}_\alpha(\alpha=1,2,\cdots,n)$ 给定时，量化区间的端点值 $x_\alpha(\alpha=1,2,\cdots,n-1)$ 必须是量化值 $\hat{x}_\alpha(\alpha=1,2,\cdots,n)$ 中两个邻近点的中点值。同样，在设计最佳矢量量化器时，重要的问题是如何划分量化区间和确定量化矢量。Gray 等人把标量量化中设计最佳量化器的两个条件，推广到设计最佳矢量量化器中。分别在两个给定条件下，寻找最佳划分与最佳码书，使平均失真最小；即一是在给定条件下，寻找信源空间的最佳划分，使平均失真最小；二是在给定划分条件下，寻找最佳码书，使平均失真最小。下面分别讨论。

① 最佳划分：由于码书已给定，因此可以用最近邻准则 NNR 得到最佳划分。图 7.4 为 $K=2$ 的最佳划分示意图。

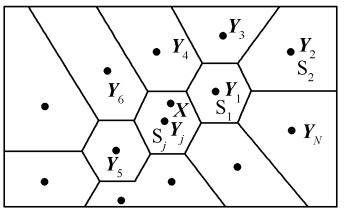


图 7.4 最佳划分示意图

信源空间 $\mathscr{X}$ 中的任一点矢量 $\boldsymbol{X}$，$\boldsymbol{X}\in S_j$（图 7.4 中所示意的是 $K=2$ 的平面），如果任意输入矢量 $\boldsymbol{X}$ 和码字 $\boldsymbol{Y}_j$ 的失真小于它和其他码字 $\boldsymbol{Y}_i\in\mathscr{Y}_N$ 的失真，即

$$S_j=\{\boldsymbol{X}\mid \boldsymbol{X}\in\mathscr{X} \text{ 且 } d(\boldsymbol{X},\boldsymbol{Y}_j)\leqslant d(\boldsymbol{X},\boldsymbol{Y}_i)\} \quad i\neq j, i\in I_N \tag{7.12}$$

则 S_j 为最佳划分。如果 $\boldsymbol{X}$ 落在边界上，可以在不增加失真的前提下，将 $\boldsymbol{X}$ 置于任何邻近区间中。由于给定码书 $\mathscr{Y}_N=\{\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_j,\cdots,\boldsymbol{Y}_N\}$ 共有 N 个码字，所以可以把信源空间划分成 N 个区间 $S_j(j=1,2,\cdots,N)$。通常把这种划分称为 Voronoi 划分，对应的子集 $S_j(j=1,2,\cdots,N)$ 称为 Voronoi 胞腔（cell），下面简称胞腔。

② 最佳码书：给定了划分 S_i（并不是最佳划分）后，为了使码书的平均失真最小，码字 $\boldsymbol{Y}_i$ 必须为相应划分 $S_i(i=1,2,\cdots,N)$ 的形心，即

$$\boldsymbol{Y}_i=\min_{Y\in R^k}{}^{-1}E[d(\boldsymbol{X},\boldsymbol{Y})\mid \boldsymbol{X}\in S_i] \tag{7.13}$$

式中，$\min^{-1}$表示选取的$\boldsymbol{Y}$是平均失真$E[d(\boldsymbol{X},\boldsymbol{Y})\mid \boldsymbol{X}\in S_i]$为最小的$\boldsymbol{Y}$。

对于一般的失真测度和信源分布，很难找到形心的计算方法，但对一些简单的分布和好的失真测度是容易找到形心的计算方法的。例如对于由训练序列定义的样点分布和常用的均方失真测度，形心就可由下式给出

$$\boldsymbol{Y}_i=\frac{1}{|S_i|}\sum_{\boldsymbol{X}\in S_i}\boldsymbol{X} \tag{7.14}$$

式中，$|S_i|$表示集合S_i中元素的个数(S_i集中有$|S_i|$个$\boldsymbol{X}$)。

有了上述的最佳划分和最佳码书两个条件，就可以得到矢量量化器的设计算法了。

7.4 矢量量化器的设计算法及 MATLAB 实现

7.4.1 LBG 算法

设计矢量量化器的主要任务是设计码书$\mathcal{Y}_N$。对于给定码字数目N的情况下，由上节所述的两个必要条件可以推导出一个矢量化器的设计算法。这个算法是由 Linde，Buzo 和 Gray 三人在 1980 年首次提出的，它是标量量化器中 Lloyd 算法的多维推广，常称为 LBG 算法。此算法既可以用于已知信源分布特性的场合，也可以用于未知信源分布特性，但要知道它的一列输出值(称为训练序列)的场合。由于对实际信源(如语声等)很难准确地得到多维概率分布，因而通常多用训练序列来设计矢量量化器。下面分别给出这两种情况下的迭代算法。

1. 算法一：已知信源分布特性的设计算法

图 7.5 所示的是已知信源分布特性的算法流程图，具体步骤如下：

① 给定初始码书$\mathcal{Y}_N^{(0)}$，即给定码书大小N和码字$\{\boldsymbol{Y}_1^{(0)},\boldsymbol{Y}_2^{(0)},\cdots,\boldsymbol{Y}_N^{(0)}\}$，并置$n=0$，设起始平均失真$D^{(-1)}\to\infty$，以及给定计算停止门限$\varepsilon$，$(0<\varepsilon<1)$。

② 用码书$\mathcal{Y}_N^{(n)}$，根据最佳划分原则构成N个胞腔$S_j^{(n)}$($j=1,2,\cdots,N$)。

③ 计算平均失真与相对失真。

平均失真为

$$D^{(n)}=E[d(\boldsymbol{X},\boldsymbol{Y})]=\sum_{i=1}^{N}P_iE[d(\boldsymbol{X},\boldsymbol{Y}_i)\mid X\in S_i^{(n)}] \tag{7.15}$$

相对失真为

$$\widetilde{D}^{(n)}=\left|\frac{D^{(n-1)}-D^{(n)}}{D^{(n)}}\right| \tag{7.16}$$

若$\widetilde{D}^{(n)}\leqslant\varepsilon$，则计算停止，此时的码书$\mathcal{Y}_N^{(n)}$就是设计好的码书$\mathcal{Y}_N=\mathcal{Y}_N^{(n)}$，否则进行第④步。

④ 利用式(7.14)计算这时划分的各胞腔的形心，由这N个新形心$\{\boldsymbol{Y}_1^{(n+1)},\boldsymbol{Y}_2^{(n+1)},\cdots,\boldsymbol{Y}_N^{(n+1)}\}$构成新的码书$\mathcal{Y}_N^{(n+1)}$并置$n=n+1$，返回第②步再进行计算，直到$\widetilde{D}^{(n+L)}\leqslant\varepsilon$，得到所要求设计的码书$\mathcal{Y}_N=\mathcal{Y}_N^{(n+L)}$为止。

2. 算法二：已知训练序列的设计算法

图 7.6 所示的已知训练序列的设计算法的流程图，具体步骤如下：

① 给定初始码书$\mathcal{Y}_N^{(0)}$，即给定码书大小N和码字$\{\boldsymbol{Y}_1^{(0)},\boldsymbol{Y}_2^{(0)},\cdots,\boldsymbol{Y}_N^{(0)}\}$并置$n=0$，设起始平均失真$D^{(-1)}\to\infty$，给定计算停止门限$\varepsilon(0<\varepsilon<1)$。

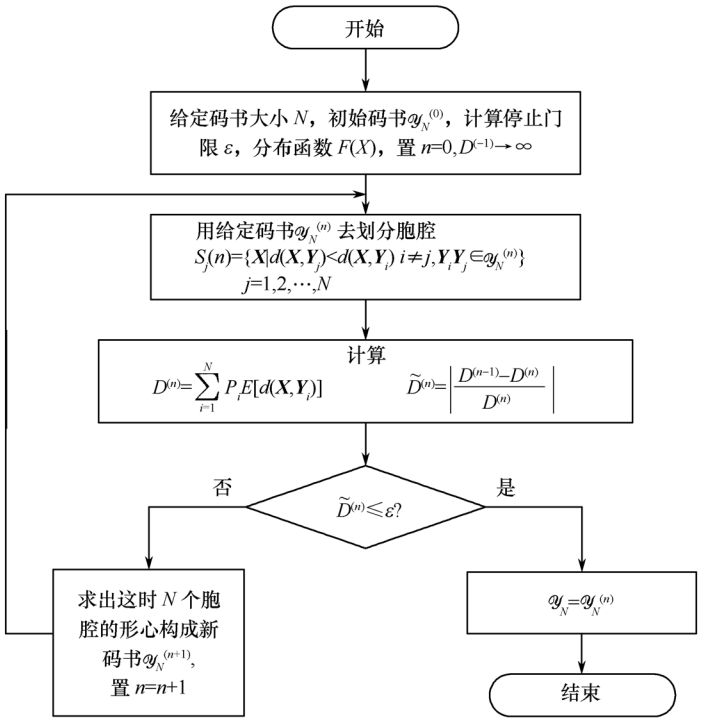


图 7.5　已知信源分布特性的算法流程图

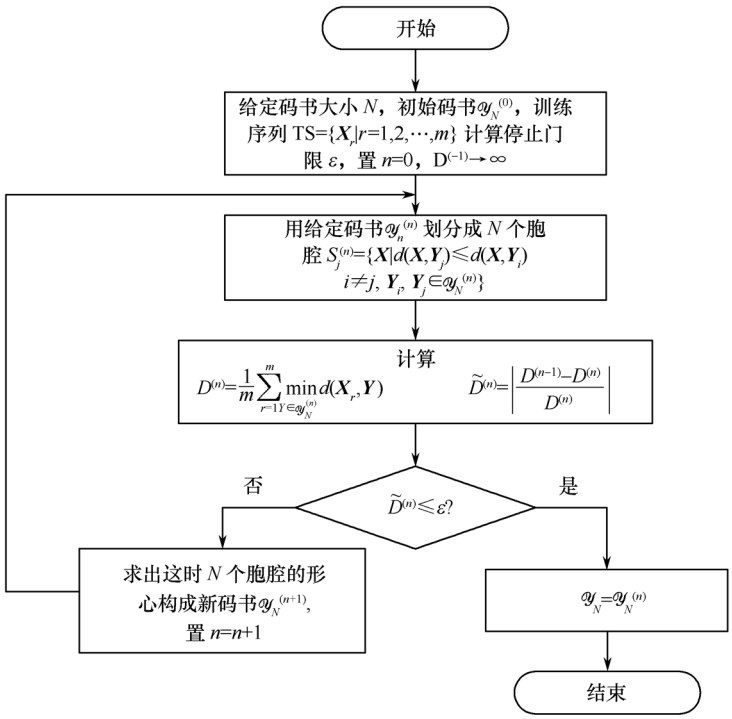


图 7.6　已知训练序列的算法

② 用码书 $\mathscr{Y}_N^{(n)}$ 为已知形心，根据最佳划分原则把训练序列 TS=$\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_m\}$ 划分为 N 个胞腔，即

$$\delta_j^{(n)}=\{\boldsymbol{X}\mid d(\boldsymbol{X},\boldsymbol{Y}_j)<d(\boldsymbol{X},\boldsymbol{Y}_i)\} \tag{7.17}$$

$$i\neq j,\boldsymbol{Y}_i,\boldsymbol{Y}_j\in\mathscr{Y}_N^{(n)},\boldsymbol{X}\in\mathrm{TS}\quad(j=1,2,\cdots,N)$$

③ 计算平均失真与相对失真

平均失真为

$$D^{(n)}=\frac{1}{m}\sum_{r=1}^{m}\min_{\boldsymbol{Y}\in\mathscr{Y}_N^{(n)}}d(\boldsymbol{X}_r,\boldsymbol{Y}) \tag{7.18}$$

式中，$\boldsymbol{X}_r\in\mathrm{TS}$；$r=1,2,\cdots,m$。

相对失真为

$$\widetilde{D}^{(n)}=\left|\frac{D^{(n-1)}-D^{(n)}}{D^{(n)}}\right| \tag{7.19}$$

若$\widetilde{D}^{(n)}\leqslant\varepsilon$，则停止计算，当前的码书 $\mathscr{Y}_N^{(n)}$ 就是设计好的码书 $\mathscr{Y}_N=\mathscr{Y}_N^{(n+L)}$，否则进行第④步。

④ 利用式(7.14)计算这时划分的各胞腔的形心，由这 N 个新形心$\{\boldsymbol{Y}_1^{(n+1)},\boldsymbol{Y}_2^{(n+1)},\cdots,\boldsymbol{Y}_N^{(n+1)}\}$构成新的码书 $\mathscr{Y}_N^{(n+1)}$，并置 $n=n+1$，返回第②步再进行计算，直到$\widetilde{D}^{(n+L)}\leqslant\varepsilon$，得到所要求的码书 $\mathscr{Y}_N=\mathscr{Y}_N^{(n+L)}$ 为止。

从理论上来讲，当训练序列充分长时，以上两种算法有某种等效性。Gray、Kieffer 和 Linde 在 1980 年证明，当信源是矢量平稳且遍历时，若训练序列长度 $m\to\infty$，算法一和算法二是等价的。1985 年，Subin 和 Gray 又把这个结果进一步推广到一大类信源的场合。除证明了极限情况下的结论外，他们还证明了对一个固定的迭代次数，算法二设计的矢量量化器逼近于算法一设计的矢量量化器。

7.4.2 初始码书的选定与空胞腔的处理

从前面讨论的两种 LBG 实际算法中可见，初始码书如何选取，对最佳码书设计是很有影响的。下面介绍两种初始码书选取方法。

(1) 随机法：这种方法是从训练序列中随机选取 N 个矢量作为初始码字，构成初始码书 $\mathscr{Y}_N^{(0)}=\{\boldsymbol{Y}_1^{(0)},\boldsymbol{Y}_2^{(0)},\cdots,\boldsymbol{Y}_N^{(0)}\}$的。此时的优点是不用初始化计算，从而可大大地减少计算时间；另外一个优点是由于初始码字选自训练序列中，因而无空胞腔问题。它的缺点是可能会选到一些非典型的矢量作为码字，因而该胞腔中只有很少矢量，甚至只有一个初始码字，而且每次迭代又都保留了这些非典型矢量或非典型矢量的形心；另外一个缺点是会造成在某些空间把胞腔分得过细，而有些空间分得太大。这两个缺点都会导致码书中有限个码字得不到充分利用，设计的矢量量化器的性能就可能较差。

(2) 分裂法：这种方法是 1980 年由 Linde、Buzo 和 Gray 提出的，具体步骤如下：

① 计算所有训练序列 TS 的形心，将此形心作为第一个码字 $\boldsymbol{Y}_1^{(0)}$。

② 用一个合适的参数 A，乘以码字 $\boldsymbol{Y}_1^{(0)}$，形成第二个码字 $\boldsymbol{Y}_2^{(0)}$。

③ 以码字 $\boldsymbol{Y}_1^{(0)}$、$\boldsymbol{Y}_2^{(0)}$ 为简单的初始码书，即

$$\mathscr{Y}_2^{(0)}=\{\boldsymbol{Y}_1^{(0)},\boldsymbol{Y}_2^{(0)}\}$$

用前面所述的 LBG 算法，去设计仅含 2 个码字的码书 $\mathscr{Y}_2^{(n)}=\{\boldsymbol{Y}_1^{(n)},\boldsymbol{Y}_2^{(n)}\}$。

④ 将码书 $\mathscr{Y}_2^{(n)}$ 中的 2 个码字 $\boldsymbol{Y}_1^{(n)}$、$\boldsymbol{Y}_2^{(n)}$ 分别乘以合适的参数 B，得到 4 个码字 $\boldsymbol{Y}_1^{(n)}$、$\boldsymbol{Y}_2^{(n)}$，

$B\boldsymbol{Y}_1^{(n)}$，$B\boldsymbol{Y}_2^{(n)}$。

⑤ 以这 4 个码矢为基础，按步骤③去构成含 4 个码字的码书，再乘以合适的参数以扩大码字的数目。如此反复，经 $\log_2 N$ 次设计，就得到所要求的有 N 个码字的初始码书 $\mathcal{Y}_N^{(0)}$。

在此方法中，这些参数的选择对初始码书的设计性能有一定影响。这些参数可选为一个固定常数，也可以选为码字的增益。用分裂法形成的初始码书，其性能较好。当然以此初始码书设计的矢量量化器的性能也较好，但是计算工作量大。

在 LBG 算法中，遇到的另一个问题是空胞腔和随机选择法中的非典型矢量如何处理。下面分别说明。

① 去细胞分裂法。首先把某空胞腔中的形心，即码字 Y_z 去掉，然后将最大的胞腔 S_M 分裂为 2 个小胞腔。分裂方法如下：

(a) 用一个合适的参数 A 去乘以原形心 Y_M，得到 2 个码字：$Y_{M1}=Y_M$，$Y_{M2}=AY_M$；

(b) 以 Y_{M1}，Y_{M2} 2 个码字来划分这个大胞腔，构成 2 个小胞腔 S_{M1}，S_{M2}。它们分别为

$$S_{M1}=\{\boldsymbol{X}\mid d(\boldsymbol{X},\boldsymbol{Y}_{M1})\leqslant d(\boldsymbol{X},\boldsymbol{Y}_{M2}),\boldsymbol{X}\in S_M\} \tag{7.20}$$

$$S_{M2}=\{\boldsymbol{X}\mid d(\boldsymbol{X},\boldsymbol{Y}_{M2})\leqslant d(\boldsymbol{X},\boldsymbol{Y}_{M1}),\boldsymbol{X}\in S_M\} \tag{7.21}$$

有时为了更精确起见，可以再计算 S_{M1}，S_{M2} 胞腔的形心，用类似于 LBG 的算法构成含 2 个码字的码书的办法，来进行分裂。此方法的优点是由于用两个小胞腔替代了 1 个大胞腔，其量化失真减小了，量化器的总失真也减小了，因此性能得到改善。

② 非典型码字的处理。在随机选择法中，存在一些非典型矢量，用它们去形成胞腔时，胞腔中往往只有少数几个矢量，甚至只有它们自己本身一个矢量。其实在别的设计算法中，也有只含很少几个矢量的胞腔，此时一般采用下面的办法来处理：

(a) 重新选择随机初始码字，直到没有非典型码字为止。

(b) 把这种胞腔中少数矢量分别归并到邻近的各个胞腔中，再用分裂法把其中一个最大的胞腔分裂为 2 个小胞腔。

7.4.3 已知训练序列的 LBG 算法的 MATLAB 实现

假设有一段语音信号命名为 lbg_7.txt，其采样频率为 16kHz，用其作为训练序列，使用 MATLAB 编程实现 LBG 算法来产生码书。初始码书从训练序列每隔 5 个样本选取一组。程序中的参数意义：codebook_size 表示码书大小；codebook_dimen 表示码书维数；训练样本个数为 signal_num，训练样本从输入数据文件选取。循环结束条件可以是循环次数到，也可以是相对失真达到指定条件。

【程序 7.1】train_codebook.m

```
clear all
codebook_size= 6;% 码书大小
codebook_dimen= 7;% 码书维数
signal_num= 100;% 参加训练样本的个数
circle_num= 20;% 码书训练循环次数，可选项，如果根据相对失真作为结束条件，就不使用
                % 该变量
fid= fopen('lbg_7.txt','rt');           % 读入数据文件 lbg_7.txt
input= fscanf(fid,'% f');% 把输入数据文件中的数据赋给 input
fclose(fid);
```

```
num= size(input/codebook_dimen);% 读输入数据大小
x= input(1000:1000+ signal_num* codebook_dimen);
% 取出输入样本文件中 1000 到 1500 共(500/codebook_dimen= 100= signal_num)组数据,
% 作为训练样本
s= zeros(codebook_size,codebook_dimen);% 初始化初始码书
train_signal= zeros(signal_num,codebook_dimen);
final_codebook= zeros(codebook_size,codebook_dimen);% 初始化最终码书
y_center= zeros(codebook_size,codebook_dimen);  % 初始化新码书质心
r= 1;
for i= 1:signal_num
    for j= 1:codebook_dimen
        train_signal(i,j)= x(r);
    r= r+ 1;
    end
end
% 选择初始码书
for i= 1:codebook_size
    for j= 1:codebook_dimen
       s(i,j)= train_signal(i* 5,j); % 每隔 5 个样本取一个样本,存入 s 数组作为初始码书
    end
end
number= zeros(signal_num,1);
D= 50000;% 起始平均失真
j2= 0;
xiangdui__distort_value= 50000;
for j1= 1:circle_num;% 让程序循环运行 circle_num 次结束
  while(xiangdui__distort_value> 0.0000001)% 当相对失真小于 0.000001 时结束程序
   j2= j2+ 1;% 如果以相对失真为循环结束条件,j2 可记录下循环次数
   % 求与训练样本距离最近的码书,则距离最近的码书索引就是训练样本所属的码书号
   for j= 1:signal_num     %  signal_num:训练样本的个数
       for k= 1:codebook_size
            A= 0;
            for m= 1:codebook_dimen
                A= A+ (train_signal(j,m)- s(k,m))^2;% 计算训练样本与当前码书质心
                                                    % 的距离
            end
           d(k)= A;
       end
       [dn,I]= min(d);%  找出训练样本与所有当前码书距离最小值及对应的码书索引
       number(j)= I;
   end % 求与训练样本距离最近的码书,则距离最近的码书索引就是训练样本所属的码书号结束
   N1= zeros(codebook_size,1);% N1:每个码书包含的样本个数
   % -----求码书质心过程------
   for t= 1:codebook_size
```

```
        y= zeros(codebook_dimen,1);  % codebook_dimen:码书维数
        N= 0;
        for j= 1:signal_num   %  signal_num:训练样本的个数
            if t= = number(j);
               for m= 1:codebook_dimen
                 y(m)= y(m)+ train_signal(j,m);
               end
                N= N+ 1;% 计算属于每个码书的样本个数
            end
        end
        N1(t,1)= N;% 属于每个码书的样本个数
        if N1(t,1)> 0
             for m= 1:codebook_dimen
             y_center(t,m)= y(m)/N1(t,1);% 求每个码书的质心
             final_codebook(t,m)= y_center(t,m);% 把训练出来的质心赋给 final_codebook
             end
        end
    end    % -----求码书质心结束------
    % -----求平均失真------
    ave_distort(j2)= 0;
    for n= 1:signal_num
        for m= 1:codebook_dimen
          ave_distort(j2)= ave_distort(j2)+ (train_signal(n,m)-  final_codebook
                           (number(n),m))^2;
            % 求所有训练样本和其所属码书质心的距离
        end
    end
    ave_distort(j2)= ave_distort(j2)/signal_num;% 计算第 j1 次循环的平均失真
    % -----求平均失真结束------

    xiangdui__distort(j2)= abs((D- ave_distort(j2))/D); % 求相对失真
    D= ave_distort(j2);
    xiangdui__distort_value= xiangdui__distort(j2);
  end
  j1= circle_num;% 当相对失真小于 0.000001 时,直接置循环次数 j1 为 circle_num 以
                 % 结束循环
end
% 把训练好的码书写到文本文件
fid= fopen('训练好的码书.txt','w');
       for t= 1:codebook_size
           for m= 1:codebook_dimen
              fprintf(fid,'% 6.2f,',final_codebook(t,m));
           end
           fprintf(fid,'\n');
```

```
        end
    fclose(fid);
```

7.5 降低复杂度的矢量量化系统

矢量量化是一种高效的数据压缩方法，但其复杂度随矢量维数成指数增长。复杂度通常包含两个方面，一是运算量，二是存储量。前面介绍的基本矢量量化系统是全搜索矢量量化器，实际应用中，人们致力于研究降低复杂度的矢量量化系统，这种研究大致朝两个方向进行，一是寻找好的快速算法；二是使码书结构化，以减小搜索量和存储量。人们已提出多种方法，这里只介绍几种典型的方法。

7.5.1 树形搜索矢量量化器

这种方法的优点是可以减少运算量，缺点是存储量有所增加且性能也有所下降。树搜索虽有二叉树和多叉树之分，但它们的原理是相同的，这里以二叉树为例说明如下。

1. 树形搜索原理

树形图是一个连通的且无环路的有向图。由图 7.7 二叉树结构图可见，以树根第一层为起点，第二层有 2 个节点($\boldsymbol{Y}_0$,$\boldsymbol{Y}_1$)；第三层有 4 个节点($\boldsymbol{Y}_{00}$,$\boldsymbol{Y}_{01}$,$\boldsymbol{Y}_{10}$,Y_{11})；第四层(此树的最后一层)有 8 个节点，各层上的节点又称为树叶。

在进行矢量量化编码时，做逐层搜索，一直到最后一层，编码时的走步控制原则为

$$\text{控制逻辑值}=\begin{cases}0 & \text{当上子树的节点失真最小时}\\ 1 & \text{当下子树的节点失真最小时}\end{cases}$$

具体量化步骤如下：

第 1 步：分别计算输入矢量 $\boldsymbol{X}$ 与 $\boldsymbol{Y}_0$、$\boldsymbol{Y}_1$ 的失真 $d(\boldsymbol{X},\boldsymbol{Y}_0)$ 和 $d(\boldsymbol{X},\boldsymbol{Y}_1)$ 并且比较它们的大小。若 $d(\boldsymbol{X},\boldsymbol{Y}_0)>d(\boldsymbol{X},\boldsymbol{Y}_1)$，则走下支路(下子树)，到了节点 $\boldsymbol{Y}_1$ 处送出 1 码至信道；若 $d(\boldsymbol{X},\boldsymbol{Y}_0)<d(\boldsymbol{X},\boldsymbol{Y}_1)$，则走上支路(上子树)，到了节点 $\boldsymbol{Y}_0$ 处，就送出 0 码至信道。

第 2 步：若上一步走的是下支路，那么在节点 $\boldsymbol{Y}_1$ 处，再计算输入矢量 $\boldsymbol{X}$ 与节点 $\boldsymbol{Y}_{10}$、$\boldsymbol{Y}_{11}$ 的失真 $d(\boldsymbol{X},\boldsymbol{Y}_{10})$ 和 $d(\boldsymbol{X},\boldsymbol{Y}_{11})$，并且比较它们的大小。若 $d(\boldsymbol{X},\boldsymbol{Y}_{10})<d(\boldsymbol{X},\boldsymbol{Y}_{11})$，则走上支路，到 $\boldsymbol{Y}_{10}$ 处送出 0 码至信道；反之，就走下支路，到了 $\boldsymbol{Y}_{11}$ 处，送出 1 码至信道。

第 3 步：若刚才走的是上支路，那么在节点 $\boldsymbol{Y}_{10}$ 处分别计算失真 $d(\boldsymbol{X},\boldsymbol{Y}_{100})$ 和 $d(\boldsymbol{X},\boldsymbol{Y}_{101})$，并且比较它们的大小，若 $d(\boldsymbol{X},\boldsymbol{Y}_{100})>d(\boldsymbol{X},\boldsymbol{Y}_{101})$，则走下支路，到了树叶 $\boldsymbol{Y}_{101}$ 处送出 1 码到信道。$\boldsymbol{Y}_{101}$ 便是输入矢量 $\boldsymbol{X}$ 的量化矢量，在信道中传输的符号是 101。反之则走上支路，到了树叶 $\boldsymbol{Y}_{100}$ 处，送出 0 码到信道。$\boldsymbol{Y}_{100}$ 便是 $\boldsymbol{X}$ 的量化矢量，在信道中传输的是符号 100。

设二叉树码书大小 $M=2^k$，k 为正整数。在形成二叉树码书时，分裂 k 次后即可得 $M=2^k$ 个码字。图 7.7 给出的是 $M=8=2^3$ 的分裂过程，每次分裂形成码书的一层，共有 $k=3$ 层。

2. 树形结构的设计

树形搜索矢量量化器的编码器是由树型码书和相应的搜索算法构成的。这种矢量量化器的特点是译码器的码书和编码器的码书不同。译码器是采用数组型码书，因为它不必用树搜索办法去寻找相应输入矢量 $\boldsymbol{X}$ 的码字，只要根据传输来的符号到数组码书中去直读即可。图 7.8是它的原理图。

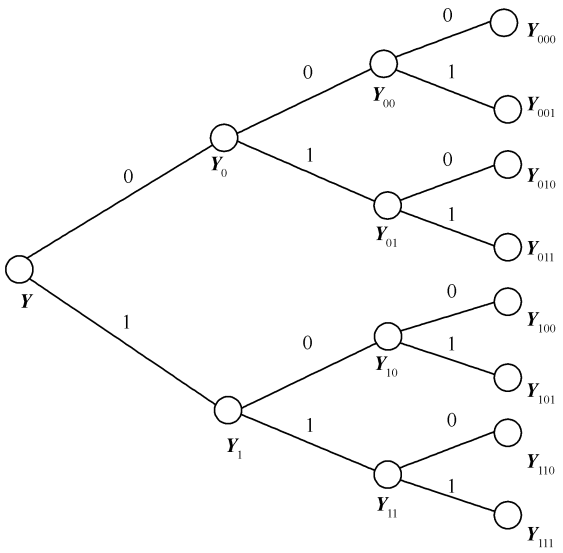


图 7.7　二叉树形结构图（ $M=8$ ）

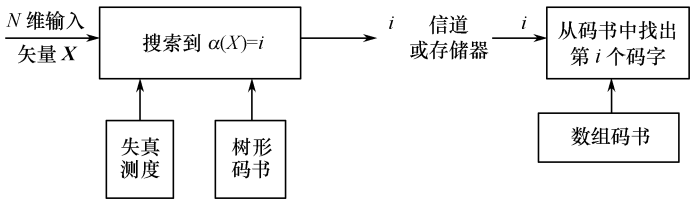


图 7.8　树形搜索矢量量化器原理框图

设计树形结构(找出各层的码字)的方法有两种:一种是从树叶开始设计;另一种是从树根开始设计。

(1) 从树叶开始设计的办法

如图 7.7 所示的四层二叉树矢量量化器,维数为 K,第四层有 $N=8$ 个码字(树叶数)。

第 1 步:假定第四层的 8 个码字,已由 LBG 设计码书的方法得到了。将这些码字,按码字距离最近配对的原则(因为是二叉树型),得到:$\{\boldsymbol{Y}_{000},\boldsymbol{Y}_{001}\}$,$\{\boldsymbol{Y}_{010},\boldsymbol{Y}_{011}\}$,$\{\boldsymbol{Y}_{100},\boldsymbol{Y}_{101}\}$,$\{\boldsymbol{Y}_{110},\boldsymbol{Y}_{111}\}$,并把它们放在相应的树叶位置上。

第 2 步:求出这些码字对的中心,如$\{\boldsymbol{Y}_{000},\boldsymbol{Y}_{001}\}$的中心为 $\boldsymbol{Y}_{00}$。总共得到四个中心:$\boldsymbol{Y}_{00}$,$\boldsymbol{Y}_{01}$,$\boldsymbol{Y}_{10}$,$\boldsymbol{Y}_{11}$,并把它们放在第三层上。

第 3 步:将第三层上的码字仍按最近距离原则配对,得到$\{\boldsymbol{Y}_{00},\boldsymbol{Y}_{01}\}$,$\{\boldsymbol{Y}_{10},\boldsymbol{Y}_{11}\}$。再求出码字对中心 $\boldsymbol{Y}_0$ 与 $\boldsymbol{Y}_1$ 并将它们放在第二层上。

这种树形码书总的尺寸为 $N_0=8+4+2=14$,即共有 14 个码字,而译码端的码字大小就是树叶数 $N=8$。

(2) 从树根开始设计的方法

同样以图 7.7 所示的四层二叉树为例,具体设计步骤如下:

第 1 步:求出整个训练序列的形心,作为初始码书。用一个合适的参数 A 去乘,得到另一个码字。而后以这两个值为初始码字,将训练序列按一定失真测度划分为两个胞腔,再计算出两个胞腔的形心 $\boldsymbol{Y}_0$ 与 $\boldsymbol{Y}_1$。用这种分裂法得到的 $\boldsymbol{Y}_0$,$\boldsymbol{Y}_1$ 便是第二层的两个码字。

第 2 步：再用上述分裂法，得到第三层的 4 个码字 $\boldsymbol{Y}_{00}$，$\boldsymbol{Y}_{01}$，$\boldsymbol{Y}_{10}$，$\boldsymbol{Y}_{11}$。这样继续下去，一直计算到树叶为止。

从上面的叙述不难看出，树搜索的过程是逐步求近似值的过程，中间的码字只起指引路线的作用。

3. 树搜索矢量量化器的复杂度

树形搜索矢量量化器的特点是以适当提高空间复杂度来降低时间复杂度。在搜索时间上，二叉树的搜索速度最快，全搜索最慢。在存储量上，二叉树多于全搜索。由于树搜索并不是从整个码书中寻找最小失真的码字，因此它的量化器并不是最佳的，也就是说树搜索矢量量化器的性能比全搜索矢量量化器的性能差。可以计算出，完成二叉树搜索所需的失真计算次数为 $2k$，失真大小比较次数为 k。全搜索时失真计算次数 2^k，失真大小比较次数为 (2^k-1) 次。当 k 值较大时，二者的差异是很大的。实际应用树搜索矢量量化器时，可以适当选择各层的树叉型数，在搜索速度、存储量及质量三者之间得到一种折中。

7.5.2 多级矢量量化器

多级矢量量化器系统由若干个普通的矢量量化器系统级联而成，如图 7.9 所示，它的第一级是一个包括 $M1$ 个码字的矢量量化器系统。对每一个输入矢量 $\boldsymbol{X}$，矢量量化编码器 1 按最近邻准则找到一个码字 $\boldsymbol{Y}_i^{(1)}$ 并计算出 $\boldsymbol{X}$ 与此码字的误差矢量 $\Delta(\boldsymbol{X},\boldsymbol{Y}_i^{(1)})$。这个误差矢量即是第二级矢量量化器系统的输入。这样一级级地推导就可以构成一个级联系统。在实际设计时，各级的码字数 $M1$、$M2$ 等一般选得大于 2。整个矢量量化编码器的输出即是各级联矢量量化编码器的输出码字的编号，而矢量量化译码器则可以根据这些编号恢复原始的输入矢量。

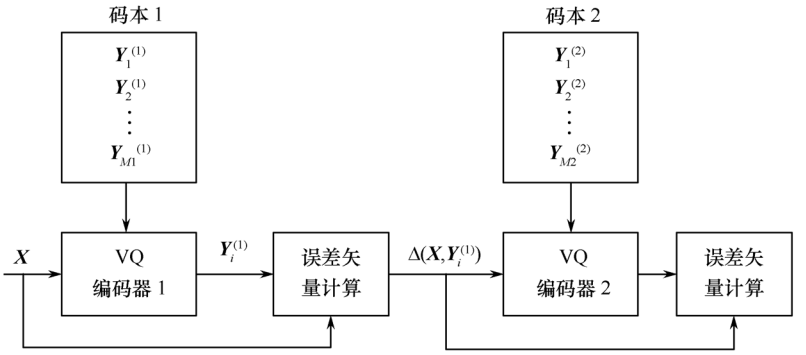


图 7.9 多级矢量量化系统编码器原理框图

多级矢量量化系统无论在减少搜索计算量方面还是减少码字存储量方面都有可观的改进，它的缺点是在同样的码书容量下，其平均量化失真大于全搜索矢量量化系统。

7.5.3 波形/增益矢量量化器

波形/增益 VQ 是一种最简单的乘积码 VQ。图 7.10 是波形/增益 VQ 实现框图。在对时域波形进行矢量量化时，可将待量化矢量的波形和增益分开，分别进行矢量量化和标量量化，这样做可以较好地改善量化性能。设输入矢量为 $\boldsymbol{X}$，其增益为 $g=\|\boldsymbol{X}\|$，$g\geqslant 0$，具有非零增益的矢量的波形为 $\boldsymbol{S}=\boldsymbol{X}/g$。可见，波形码书中所有的波形矢量均为单位增益。

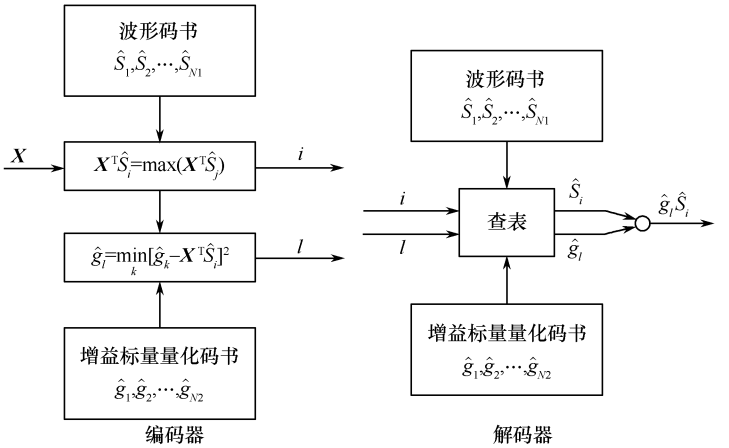


图 7.10　波形/增益矢量量化器原理框图

采用平方误差失真测度，则输入矢量和量化矢量间的失真为

$$d(\boldsymbol{X},\hat{g}\hat{\boldsymbol{S}})=\|\boldsymbol{X}-\hat{g}\hat{\boldsymbol{S}}\|^2=\|\boldsymbol{X}\|^2+\hat{g}^2-2\hat{g}\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}}=\|\boldsymbol{X}\|^2+(\hat{g}-\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}})^2-(\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}})^2 \tag{7.22}$$

式中，$\hat{g}$ 和 $\hat{\boldsymbol{S}}$ 分别是增益和波形矢量 $\boldsymbol{S}$ 的量化结果。

VQ 编码可分两步使式(7.22)达到最小。首先在 VQ 码书中找到一个码字 $\hat{\boldsymbol{S}}$，使其与输入矢量 $\boldsymbol{X}$ 的点积 $\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}}$ 达到最大值；然后在增益标量量化码书中寻找一个与 $\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}}$ 最为接近的增益值 $\hat{g}$[即使 $(\hat{g}-\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}})^2$ 达到最小]。将 $\hat{g}$ 和 $\hat{\boldsymbol{S}}$ 对应的编号传到解码器中。后者通过查表将 $\hat{g}\hat{\boldsymbol{S}}$ 作为解码输出。

波形/增益 VQ 系统在时间复杂度和空间复杂度上明显优于全搜索矢量量化器，但由于码书在相同条件下比全搜索码书差，因此性能是次优的。

7.5.4　分离均值矢量量化器

分离均值矢量量化器先将输入矢量的平均值分离出来，以较低的速率对均值进行标量量化，然后对去掉均值的输入矢量进行矢量量化。其码书的设计过程可简单描述如下：

第 1 步：根据原始训练序列计算矢量均值，对均值矢量选择合适的标量量化方法进行量化。

第 2 步：从原始训练序列矢量中减去对应矢量的量化均值，形成残差训练序列，使用 LBG 算法对该序列进行训练求得残差码书。分离均值矢量量化器同波形/增益量化器一样，通过降低量化矢量的动态范围来降低量化器的复杂性，但它的性能也比全搜索码书差。

7.5.5　有记忆的矢量量化

前面介绍的矢量量化系统都属于无记忆的矢量量化情况。因为在量化每一矢量时，都不依赖于此矢量之前或之后的任何矢量。与之相反，如果能利用过去的输入矢量的信息，来决定当前的输入矢量应该用哪一个码书进行比较，那么，通过机器的“记忆”，人们就可以利用矢量之间的相关性，来提高矢量量化的性能。有记忆的矢量量化又称反馈型的矢量量化，它是多码书的矢量量化系统。人们已经研究过多种形式的有记忆的矢量量化，这里只介绍其中的一种：自适应预测矢量量化(APVQ)。

APVQ 的工作过程如下：将输入语音信号序列分帧，构成矢量序列，对某一输入矢量 $\boldsymbol{X}_n$，用线性预测原理得到一个预测矢量 $\widetilde{\boldsymbol{X}}_n$，相减之后得到误差矢量 $\boldsymbol{e}_n=\boldsymbol{X}_n-\widetilde{\boldsymbol{X}}_n$，对此误差矢量，用 e_n 码书对它进行矢量量化，送给信道的是该量化误差矢量的下标。另一方面，还采用自适应技术，根据语音流各段的不同的统计特性，将输入矢量划分为不同类型，用不同的码书来量化，也就是由帧分类器输出附加信息，决定用哪一个码书进行误差矢量的量化和用哪一个预测器来得到预测矢量，同时这个信息也由信道传送到接收端，图 7.11 所示即为 APVQ 系统的框图。

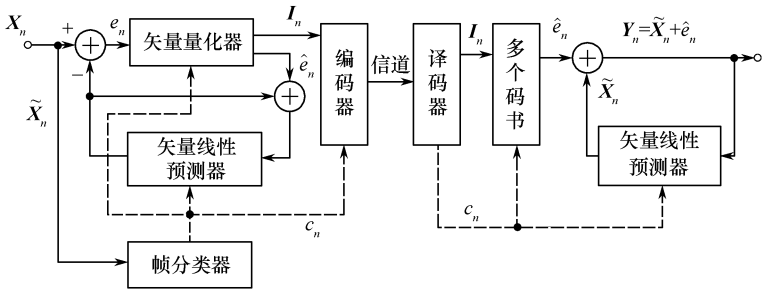


图 7.11　APVQ 系统框图

实践表明，由于 APVQ 去掉了矢量与矢量之间的编码冗余度，并且利用了语音信号的局部特性，因此，尽管复杂度比普通的矢量量化器增大了，但可以提高约 7dB 的信噪比。

第8章 语音编码

8.1 概 述

语音编码是语音信号处理的一个分支，主要用于通信领域。语音信号的数字化传输一直是通信发展的主要方向之一，语音的数字通信与模拟通信相比，无疑具有更好的效率和性能，这主要体现在：具有更好的语音质量；具有更强的抗干扰性，并易于进行加密；可节省带宽，能够更有效地利用网络资源；更加易于存储和处理。最简单的数字化方法是直接对语音信号进行模/数转换，只要满足一定的采样率和量化要求，就能够得到高质量的数字语音。但这时语音的数据量仍旧非常大，因此在进行传输和存储之前，往往要对其进行压缩处理，以减少其传输码率或存储量，即进行压缩编码。传输码率也称为数码率或编码速率，表示传输每秒钟语音信号所需的比特数。语音编码的目的就是要在保证语音音质和可懂度的条件下，采用尽可能少的比特数来表示语音。通常所说的“语音编码”，是特指通信传输系统中代表口语发声的300～3400Hz的信号。本章以前面学习过的语音信号处理技术和方法为基础，介绍语音编码基本原理和常用的编码方法。

8.2 语音编码的分类及特性

语音编码按编码方式大致可以分为三种：波形编码、参数编码和混合编码。波形编码是将时间域或变换域信号直接变换为数字信号，力求使重建语音波形保持原始语音信号的波形形状。参数编码又称声码器编码，它是将信源信号在频域或其他变换域提取特征参数，然后对这些特征参数进行编码和传输，在译码端再将接收到的数字信号译成特征参数，根据这些特征参数重建语音信号。混合编码将波形编码和参数编码结合起来，克服了波形编码和参数编码的缺点，吸收了它们的长处，能够在较低速率上得到高质量的合成语音。

8.2.1 波形编码

波形编码是降低量化每个语音样点的比特数，同时保持相对好的语音质量，在波形编码中要求重建语音信号$\hat{s}(n)$的各个样本尽可能地接近原始语音信号$s(n)$的样本值，如果令$e(n)=s(n)-\hat{s}(n)$表示量化误差或重构误差，那么波形编码的目的是在给定的传输比特率下，使误差序列$e(n)$的能量最小。传统的波形编码方法有脉冲编码调制(PCM)、自适应增量调制(ADM)和自适应差分脉冲编码调制(ADPCM)等。针对语音信号幅度分布不均匀的特点，PCM中用μ-律或A-律对信号抽样进行不均匀量化，需要用64kbit/s码率实现；ADM中对信号增量进行自适应量化，需要用32～16kbit/s码率实现；ADPCM利用波形样点之间的短时相关性，进行短时预测，对预测值与原始语音的差值(预测残差)进行编码，用32kbit/s码率可以再现高质量语音。波形编码具有语音质量好、适应能力强、算法简单、易于实现、抗噪性能强等优点。其缺点是所需的编码速率高，一般在16～64kbit/s之间。

8.2.2 参数编码

参数编码是以语音信号产生的数字模型为基础，对数字语音信号进行分析，提出一组特征参数（主要是指表征声门振动的激励参数和表征声道特性的声道参数），这些参数携带有语音信号的主要信息，编码它们只需要较少的比特数，在解码后可以由这些参数重新合成语音信号。码率的降低主要取决于分析和提取什么样的特征参数以及合成器的类型。这种编码方法力图使重建语音信号具有尽可能高的可懂度，但重建语音信号与原始语音信号样本之间没有一一对应关系，因而合成语音的音质好坏需要借助于主观评定，而缺少客观的评定标准。共振峰声码器、线性预测声码器、余弦声码器都属于参数编码器。参数编码的优点是可实现低速率语音编码，其编码速率可低至 2.4kbit/s 以下，其缺点是语音质量差，自然度较低。这类编码器对讲话环境噪声较敏感，需要安静环境才能给出较高的可懂度。

8.2.3 混合编码

波形编码虽然能够得到很好的语音质量，但它的编码速率很高，而参数编码虽然能获得很低的编码速率，但其合成语音质量不高。混合编码在保留参数编码的技术精华的基础上，引用波形编码准则去优化激励源信号，克服了原有波形和参数编码的弱点，而吸取了它们各自的长处，在 4～16kbit/s 的速率上能够合成高质量语音。多脉冲激励线性预测编码（MPE-LPC）、码激励线性预测编码（CELP）等都属于这类混合编码器。混合编码器以复杂的算法和很大的运算量为代价，在中低速率语音编码上获得了高质语音。

8.2.4 语音压缩编码的依据

一般来讲，语音编码的目的是在给定的编码速率下，使得编解码后恢复出的重构语音的质量尽可能高。提高语音编码效率的基本途径在于充分利用语音信号中的冗余度和人耳的听觉特性。

语音的冗余度主要来源于两个方面，语音信号幅度分布的非均匀性和语音样点之间的相关性。

语音信号的幅度统计特性同信号的带宽、采样时的声学条件以及统计的时间长度都有关系。对于这样一种随机过程，它的概率密度分布不满足任何一种固定的分布，它是具有动态的、时变的、多维的暂态概率密度分布的随机过程。随着统计时间长度不同，它表现的概率密度分布形式不同，一般认为长时（几十秒以上）统计幅度特性接近于 gamma 分布，而短时（几到几十毫秒）统计所得幅度特性接近于高斯分布。但无论长时或短时统计，语音信号总是小幅度出现的概率大，而大幅度出现的概率小。非均匀标量量化就是直接利用了语音信号的这一特点，使量化质量得到提高。参数编码质量的提高的理论基础也依赖于对语音信号的统计特性的进一步研究。

语音样点之间存在相关性是语音信号具有冗余度的另一原因。通过对语音发声的机理进行研究，我们知道语音是由肺呼出的气流通过声门形成的激励信号激励声道，再经唇、口和鼻辐射出来的。从信号处理的角度出发，可以把语音看成是由白噪声或周期脉冲激励信号通过一个有色滤波器所产生的。这一过程在时域上看，相当于使样点之间产生了相关性；从频域看，相当于给频谱加色，使原来的白色谱变成了非平坦的有色谱。此外，在语音中的浊音段，信号具有准周期的特性，其频谱含有谱线结构。因此，除了谱包络代表的短时相关性外，浊音段

还有长时相关性。利用语音信号的这些相关性，在时域上采用短时和长时预测，在频域上采用谱平整方法，都可以达到压缩编码比特率的目的。

语音压缩编码的第二个途径是利用人耳的听觉特性。人类听觉有一个特点，就是“听觉掩蔽效应”：一个强音能抑制一个同时存在的弱音的听觉。利用这一性质就可以抑制与信号同时存在的量化噪声，如“噪声谱形变技术”；另外，人的听觉对低频端比较敏感，而对高频端不太敏感，因此引出了“子带编码技术”；还有人的听觉对信号的相位特性不敏感，线性预测声码器利用这一特点，并不传送语音谱的相位信息，使码率能降至 2.4kbit/s 以下，仍保持高的可懂度；感觉加权滤波器和后滤波技术则利用幅度谱的适度失真来降低量化噪声对语音质量的影响。

8.3 语音编码性能的评价指标

语音编码的根本目标就是在尽可能低的编码速率条件下，重建得到尽可能高的语音合成质量，同时还应尽量减小编解码延时和算法复杂度，因此编码速率、语音质量评价、编解码延时以及算法复杂度这四个因素自然就成了评价一个语音编码算法性能的基本指标，这四个因素之间有着密切的联系，在具体评价一种语音编码算法的优劣时，需要根据具体的实际情况，综合考虑 4 个因素进行性能评价。

8.3.1 编码速率

编码速率直接反映了语音编码对语音信息的压缩程度。编码速率可以用“比特/秒”(bit/s)来度量，它代表编码的总速率，一般用 I 表示；也可以用“比特/样点”(bit/p)来度量，它代表平均每个语音样点编码时所用的比特数，用 R 表示。两者之间可以用公式 $I=R\cdot f_s$ 互相转换，其中 f_s 为抽样频率。显然，平均每样点比特数 R 越高，语音波形或参数量化则越精细，语音质量也就越容易提高，相应地对传输带宽或存储容量的要求也就越高。

降低编码速率往往是语音编码的首要目标，它直接关系到传输资源的有效利用和网络容量的提高。根据编码速率和输入语音的关系可将编码器分成两类：固定速率编码器和可变速率编码器。

现在大部分编码标准都是固定速率编码，其范围为 0.8～64kbit/s。其中，保密电话的编码速率最低，为 0.8～4.8kbit/s，其原因是它的通信信道带宽限定在 4.8kbit/s 以下。数字蜂窝移动电话和卫星电话编码器的编码速率为 3.3～13kbit/s，它使数字蜂窝系统的容量可以达到模拟系统的 3～5 倍。需要注意的是，蜂窝系统中常伴有信道编码，使总的编码速率达到 20～30kbit/s。普通电话网的编码速率为 16～64kbit/s。其中有一类特别的编码器称为宽带编码器，其编码速率为 48/56/64kbit/s，用于传送 50Hz～7kHz 的高质量音频信号，如会议电视系统。在固定速率的编码器中，有些编码器采用一些特殊的技术，以提高信道利用率，例如，语音插空技术利用语音之间的自然停顿传送另一路语音或数据。

可变速率编码是近年来出现的新技术。根据统计，两方通话大约只有 40％的时间是真正有声音的，因此一个自然的想法是采用通、断状态编码。通状态对应有声期，采用固定编码速率；断状态对应无声期，传送极低速率信息(如背景噪声特征等)，甚至不传送任何信息。更复杂的多状态编码还可以根据网络负荷、剩余存储容量等外部因素调节其码率。可变速率编码主要包括两个算法：一是语音激活检测(VAD)，主要用于确定输入信号是语音还是背景噪声，其难点在于正确识别出语音段的开始点，确保语音的可懂度；二是舒适噪声的生成(CNG)，主

要用于接收端重建背景噪声，其设计必须保证发送端和接收端的同步。

8.3.2 编码质量

语音编码质量评价可以说是语音编码性能的最根本指标，评价语音质量的方法归纳起来可以分为两类：主观评价方法和客观评价方法。

1. 语音质量主观评价方法

主观评价方法符合人听话时对语音质量的感觉，目前得到了广泛应用。主观评价方法是在一组测试者对原始语音和合成语音进行对比试听的基础上，根据某种事先约定的尺度来对语音质量划分等级。常用的方法有平均意见得分 MOS，判断韵字测试 DRT 得分和判断满意度测量 DAM 得分。国际上应用最广的是平均意见得分评定法，一般称为 MOS 评分。表 8.1 中列出了 MOS 判分标准及相应的语音质量级别。

表 8.1 MOS 分五级标准及对应语音质量

MOS 分	质量级别	失真级别
5	优	不觉察
4	良	刚有觉察
3	中	有觉察且稍觉可厌
2	差	明显觉察且可厌但可忍受
1	坏	不可忍受

在数字通信中，通常认为 MOS 分 4.0～4.5 分为高质量数字语音，达到长途电话网的质量要求，也常称之为网络质量。MOS 评分在 3.5 分左右时称为通信质量，这时能感觉到重建语音质量有所下降，但不妨碍正常通话，可以满足多数语音通信系统使用要求。MOS 评分在 3.0 分以下的常称合成语音质量，这是指一些声码器合成的语音所能达到的质量，它一般具有足够高的可懂度，但自然度及讲话人的确认等方面不够好。

虽然主观评价方法符合人类听话时对语音质量的感觉，但由于其测试结果的获得依赖于测听者个人的主观感受，因此为了减少个人反应的随意性和不可重复性，一般对测试所用的设备、数据、测试条件及测试人员都有严格的要求，并有烦琐的测听程序规定，需要消耗大量的时间、人力和费用，而且即使如此，测试结果仍然存在着一定的不可重复性，在完全相同的测试条件下重复测试，结果也会有一定的随机波动，所以主观评价方法一般都是由较大的通信组织来完成，个人很少采用。

2. 语音质量客观评价方法

客观评价方法是用客观测量的手段来评价语音编码质量，它是建立在原始语音和合成语音的数学对比之上的，常用的方法可分为时域客观评价和频域客观评价两大类。时域客观评价常用的方法有信噪比、加权信噪比、平均分段信噪比等；频域客观评价常用的方法有巴克谱失真测度 BSD 和 MEL 谱测度等。这些评价方法的特点是计算简单、结果客观、不受个人主观因素的影响，但其缺陷也很明显，就是不能完全反映人类对语音的听觉效果。虽然如平均分段信噪比、巴克谱失真测度等考虑了人耳的多种听觉特性，并做了相应的加权校正，在评价速率较高的波形编码算法时和人的主观感觉比较符合，但在参数编码算法和混合编码算法的评价中仍然存在着上述问题。

分段信噪比采用分段(10～30ms)的方法来分别计算每一段语音信号的信噪比,因此能够反映出量化器对不同电平输入段的量化质量。设 $s_m(i)$ 为第 m 段的输入语音信号,$\hat{s}_m(i)$ 为第 m 段的合成语音信号,每段中有 M 个语音样点,则第 m 段的语音分段信噪比定义为

$$\mathrm{SNR}_{\mathrm{seg}}(m) = 10\lg\left[\frac{\sum_{i=1}^{M} s_m^2(i)}{\sum_{i=1}^{M}(s_m(i) - \hat{s}_m(i))^2}\right] \quad (\mathrm{dB}) \tag{8.1}$$

如果输入语音共有 N 段,平均分段信噪比为

$$\mathrm{SNR}_{\mathrm{aseg}} = \frac{1}{N}\sum_{m=1}^{N}\mathrm{SNR}_{\mathrm{seg}}(m) \quad (\mathrm{dB}) \tag{8.2}$$

3. PESQ 语音质量评价法

ITU-T 从 1996 年开始进行了一系列的实验来寻找一种新的语音质量评价模型,以期能适应更广泛的编解码器和网络情况,具有更好的性能和表现。2001 年 2 月,在通过了由 9 种语言、在不同的真实和仿真的网络中采集语音构成大规模样本库的全面测试评价后,感知语音质量评价 PESQ 方法被 ITU-T 确定为 P.862 建议,成为了窄带电话网络和语音编解码器的端到端语音质量的客观评价方法。

ITU-T 的 P.862 建议提供了(－0.5,4.5)内的原始输出评分 PESQ 值,同时又给出一个“映射函数”将 P.862 的输出结果转换成一个 MOS-LQO 评分,以便于将 P.862 的结果和 MOS 的结果进行线性比较。PESQ 算法将语音的频率、响度等物理特性与人类心理上的感知特性的关系通过数学模型对应起来,用客观模型来模拟主观感觉的评价。该模型采用时频映射、频率弯折等方法,结合感知模型,将语音中“可感知”的特性在数学上尽可能完美的表达。PESQ 具有广泛的适用性,具有端到端的复杂信道和网络语音质量评价能力,适用于移动通信系统在内的通信网络的语音通信质量评价。

8.3.3 编解码延时

编解码延时一般用单次编解码所需的时间来表示,在实时语音通信系统中,语音编解码延时同线路传输延时的作用一样,对系统的通信质量有很大影响。过长的语音延时会使通信双方产生交谈困难,而且会产生明显的回声而干扰人的正常思维。因此,在实时语音通信系统中,必须对语音编解码算法的编解码延时提出一定的要求。对于公用电话网,编解码延时通常要求不超过 5～10ms,而对于移动蜂窝通信系统,允许最大延时不超过 100ms。延时影响通话质量的另一个原因是回声。当延时较小时,回声同话机侧音及房间交混回响声相混,因而感觉不到。但当往返总延时约 100ms,发话者就能从手机中听到自己的回声,从而影响通话质量。

8.3.4 算法复杂度

算法复杂度主要影响到语音编解码器的硬件实现,它决定了硬件实现的复杂程度、体积、功耗及成本等。对一些复杂的语音编码算法,一般编码算法的复杂程度与语音质量有密切关系。在同样速率的情况下,复杂一些的算法将会获得更好一些的语音质量。算法的复杂程度与硬件实时实现也有密切关系。它对数字信号处理芯片的运算能力及存储器容量都有一定的要求。运算能力可用处理每秒钟信号样本所需的数字信号处理器(DSP)指令条数来衡量其计

算复杂度，用单位“百万次操作/秒”MOPS 或“百万条指令/秒”MIPS 等来对算法复杂度进行描述。存储器容量通常用千字节(KB)的数量来衡量。算法越复杂则运算量越大，需要一片或多片 DSP 芯片以及较大容量的存储区方可实现。

8.4 语音信号波形编码

8.4.1 脉冲编码调制 PCM

1. 均匀量化 PCM

脉冲编码调制是最简单的波形编码方法，它把语音信号样本幅值量化到 $N=2^B$ 个码字中的一个，这样每个样本需用 B 比特来表示。假定信号带宽是 WHz，根据取样定理，总的比特率(每秒钟比特数)将是 $2WB$ 比特/秒。均匀量化 PCM 和普通的 A/D 变换是完全相同的，它没有利用语音信号的任何性质，也没有进行压缩。这种编码方法中，输入信号 $x(n)$ 幅值的范围被分成 N 个相同宽度的区间，所有落入同一区间的样本都编码成相同的二进制码字。语音是非平稳随机信号，电话语音电平变化超过 40dB。对小信号电平输入，信噪比应保证约 20～30dB，即最大信噪比应为 60～70dB。只要 N 足够大，我们可以合理地假定，量化误差 $e(n)$ 在各个宽度为 Δ 的区间里是均匀分布的，信号对量化噪声的功率比(简称信噪比)可近似地写成

$$\mathrm{SNR}=\sigma_x^2/\sigma_e^2=\sigma_x^2/(\Delta^2/12) \tag{8.3}$$

或用分贝表示时，有

$$\mathrm{SNR(dB)}=6.02B+4.77-20\log(X_{\max}/\sigma_x) \tag{8.4}$$

式中，σ_x^2 和 σ_e^2 是输入信号和量化噪声的方差或平均能量，$X_{\max}$ 是输入信号的峰值，B 是量化的比特数。进一步假定，输入量化器的信号值范围限制在 $-4\sigma_x$～$+4\sigma_x$，即 $X_{\max}=4\sigma_x$，那么有

$$\mathrm{SNR(dB)}=6.02B-7.2 \tag{8.5}$$

这表明量化器每增加一个比特，信号量化噪声比增加 6dB。量化比特数 B 的选择要考虑到输入信号已有的信噪比。当要求 60dB 的 SNR 时，B 至少应取 11。此时，对于带宽为 4kHz 的电话语音信号，若采样率为 8kHz，则 PCM 要求的速率为 88kbit/s。这样的比特率是比较高的。

均匀量化 PCM 在下列两个假设条件下效果是很好的：①输入信号幅度变化范围是已知的；②信号幅度值在已知的范围内是均匀分布的。然而，语音信号是一个非平稳的过程，最强的音和最弱的音之间相差 30dB 以上。并且不同的人、不同场合、讲话响、轻相差甚远。因此均匀量化要求的两个条件对语音信号来讲实际上都不可能满足。如果我们设计的量化器动态范围太小，那么当输入语音信号幅度超过这个范围时，会出现过载噪声或者饱和噪声；反之，设计的量化器动态范围很大，那么量化间隔相应增加，量化噪声就大，有时甚至淹没一些微弱的语音。此外，从式(8.4)还可以看到，信号量化噪声比和输入信号的方差有关，若输入信号方差只有量化器设计范围的一半，则信噪比下降 6dB。显然一个清音段的方差也许比浊音段的方差要低 30dB，那么短时信噪比在清音段期间要比浊音段期间低得多，因此为了在均匀量化时保持听觉上满意的效果，不得不使用较多的量化比特数，而这又是不现实的，所以，必须研究更高效的编码方案。

2. 对数 PCM

改进 PCM 编码器性能的一个方法是采用非均匀的量化，即让量化间隔大小不相等。对小的输入信号值量化间隔较小，对大的信号值量化间隔较大。这样，可以对任何输入信号电平保持近似相同的信噪比。采用非均匀量化后，显然只要用较小的量化比特数，在满足小信号有一定的信噪比同时，又有足够的动态范围使大信号时不会出现过载问题。如果我们能够测定语音信号幅度的概率密度函数，那么对于某个给定的量化比特数，非均匀量化器完全可以设计得使量化噪声达到最小。然而实际的概率密度函数和设计的概率密度函数往往不容易匹配，这时量化器的性能会急剧降低。

我们希望量化器性能既不敏感于输入信号的方差，又不敏感于输入信号的概率密度函数，常用的 μ -律或 A-律量化器就是具有这种特性的非均匀量化器。下面对 μ-律量化器做一介绍。非均匀量化可以等效于把信号幅度非线性地压缩后再进行线性量化，从前面的分析不难看到，对数压缩是比较理想的。这一点可以简单地证明如下：假如均匀量化前，先用对数做幅度压缩，译码后用指数函数进行扩张，即

$$y(n)=\ln|x(n)| \tag{8.6}$$

其反变换

$$x(n)=\exp[y(n)]\operatorname{sgn}[x(n)] \tag{8.7}$$

式中 sgn[·]是符号函数。那么量化后有

$$\hat{y}(n)=Q[\ln|x(n)|]=\ln|x(n)|+e(n) \tag{8.8}$$

假设 $e(n)$与 $\ln|x(n)|$不相关，量化后对数幅度的反变换为

$$\hat{x}(n)=\operatorname{sgn}[x(n)]\exp[\hat{y}(n)]=|x(n)|\operatorname{sgn}[x(n)]\exp[e(n)]=x(n)\exp[e(n)] \tag{8.9}$$

当 $e(n)$很小时，上面公式近似为

$$\hat{x}(n)=x(n)[1+e(n)]=x(n)+x(n)e(n)=x(n)+f(n) \tag{8.10}$$

式中 $f(n)=x(n)e(n)$。由于 $x(n)$与 $e(n)$是统计独立的，因此有

$$\sigma_f^2=\sigma_x^2\sigma_e^2,\quad \mathrm{SNR}=\sigma_x^2/\sigma_f^2=1/\sigma_e^2 \tag{8.11}$$

这就证明了信噪比和信号方差无关，它仅取决于量化间隔。式(8.6)那样的量化器实际上是不能实现的，因为那里最大值与最小值的比假设成无限大($\ln(0)=-\infty$)，需要无限个量化单元；在实用中是将对数压缩特性作某种近似，μ -律压缩就是最常用的一种。μ -律压缩的定义是：

$$y(n)=F_\mu[x(n)]=X_{\max}\frac{\ln[1+\mu|x(n)|/X_{\max}]}{\ln(1+\mu)}\operatorname{sgn}[x(n)] \tag{8.12}$$

式中，$X_{\max}$是信号 $x(n)$的最大幅值，μ 是参变量，用来控制压缩程度，$\mu=0$ 表示没有压缩，μ 值越大压缩越厉害，故称之为 μ -律压缩。

图 8.1 给出了 μ -律压缩的输入输出特性曲线，根据这个特性曲线可知，当输入小幅度值时，等效量化间隔小，输入大幅度值时量化间隔大。

在 μ -律量化情况下，可推导出其信号量化噪声比公式为

$$\begin{aligned}\mathrm{SNR(dB)}=&6.02B+4.77-20\log[\ln(1+\mu)]\\&-10\log[1+(X_{\max}/\mu\sigma_\chi)^2+\sqrt{2}(X_{\max}/\mu\sigma_\chi)]\end{aligned} \tag{8.13}$$

将此结果与式(8.4)比较可见，SNR 值与量($X_{\max}/\sigma_\chi$)的依赖关系要松得多，当 μ 增大时，SNR 对($X_{\max}/\sigma_\chi$)的变化越来越不敏感。

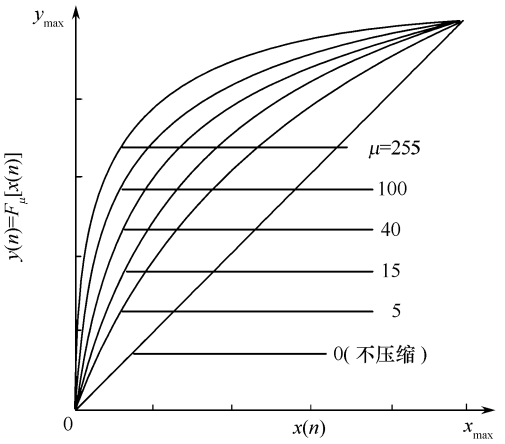


图 8.1　μ-律特性的输入输出结果

与 μ-律量化具有相同效果的还有 A-律量化，A-律压缩特性可表示成

$$F_A[x(n)]=\begin{cases}\dfrac{A|x(n)|/X_{\max}}{1+\ln A}\operatorname{sgn}[x(n)], & 0\leqslant|x(n)|/X_{\max}\leqslant 1/A\\ \dfrac{1+\ln(A|x(n)|/X_{\max})}{1+\ln A}\operatorname{sgn}[x(n)], & 1/A\leqslant|x(n)|/X_{\max}\leqslant 1\end{cases}\tag{8.14}$$

和 μ-律比较，A-律压缩的动态范围略小些，在小信号时质量要较 μ-律要差些。A-律最小量化间隔是 2/4096，而 μ-律是 2/8159，事实上这二者的差别是不易觉察到的。无论是 A-律或 μ-律，其特性在 x 值小时都是线性的，在 x 值大时则呈现对数压缩特性。

采用 A-律或 μ-律量化的脉冲编码调制系统统称为对数 PCM 系统，是目前最为成熟的一种语音压缩编码方法。8 比特的对数 PCM(64kbit/s)于 1972 年被 ITU-T 制定为 G.711 标准，已普遍地应用于数字电话系统中。不同国家和地区的体制不同，在北美和日本 PCM 标准是采用 $\mu=255$ 的 μ-律 PCM，欧洲 PCM 标准则采用 $A=87.56$ 的 A-律 PCM，我国也采用 A-律。标准 μ-律或 A-律 PCM 编码器芯片早已问世，例如美国 TI 公司的 TCM2916、TCM2917、MOTOROLA 公司的 MC14403、MC14405 等，它们都是 μ-律或 A-律的单片对数 PCM 编解码器，并且内含编解码所需的滤波器。

3. 自适应量化 PCM

自适应量化是指量化器的特性自适应于输入信号的幅度的变化，即一个自适应量化器的量化间隔应自适应地改变，并与输入信号的幅度方差保持相匹配，或者等效地在一个固定的量化器前，加一个自适应的增益控制，使进入量化器的输入信号方差保持为固定的常数。采用自适应量化器的 PCM 就称为“自适应脉冲编码调制”APCM。

图 8.2 是这两种 APCM 方法的框图，这两种方法中，都需要随时估计输入信号的时变幅值，以修正量化间隔 $\Delta(n)$或增益 $G(n)$的值。图中上标“′”表示接收端得到的参量，如果传输信道没有引入误码，那么有 $c'(n)=c(n)$，$\Delta'(n)=\Delta(n)$，$G'(n)=G(n)$等。关于自适应的速度，如果是每个样本或者几个样本进行自适应调整，称为“瞬时自适应”；如果是较长时间才进行自适应调整的，例如浊音与清音的幅值往往相差很大，但在浊音期间或清音期间幅度方差基本保持不变，那么这时的自适应可称为“音节自适应”。根据 $\Delta(n)$和 $G(n)$的估计方法不同，自适应方案又可分为“前馈自适应”和“反馈自适应”两种。

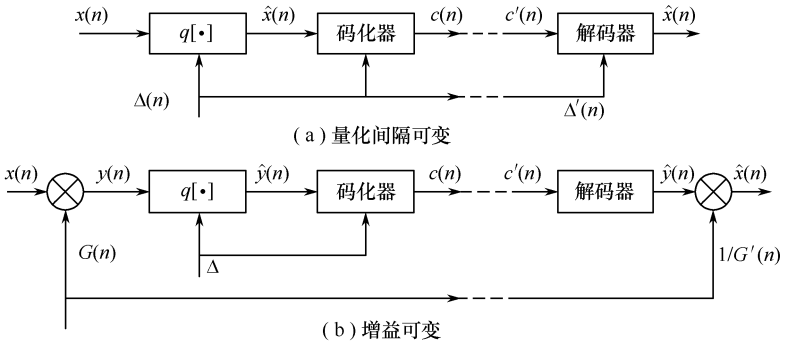


图 8.2　自适应量化框图

(1) 前馈自适应

所谓前馈自适应是指信号 $x(n)$的能量或方差是由输入信号 $x(n)$本身估算出来的，一般是先估算出 $x(n)$的方差 $\sigma^2(n)$后，令两种系统输出为

$$\Delta(n)=\Delta_0\sigma(n),\quad G(n)=G_0/\sigma(n) \tag{8.15}$$

即 $\Delta(n)$正比于 $\sigma(n)$，$G(n)$反比于 $\sigma(n)$，它们除了在发送端使用外，还作为边信息，随同语音样本码值一起传送到接收端去。通常认为，时变方差 $\sigma^2(n)$正比于语音信号的短时能量，而我们知道，短时能量可定义为 $x(n)$经低通滤波器 $h(n)$后的输出，因此有

$$\sigma^2(n)=\sum_{m=-\infty}^{+\infty}x^2(m)h(n-m) \tag{8.16}$$

式中，$h(n)$为低通滤波器的单位冲激响应，可由采用的窗函数求出。例如，设窗函数为

$$h(n)=\begin{cases}\alpha^{n-1}, & n\geqslant 1\\ 0, & \text{其他}\end{cases} \tag{8.17}$$

则

$$\sigma^2(n)=\sum_{m=-\infty}^{+\infty}x^2(m)\alpha^{n-m-1} \tag{8.18}$$

显然，$\sigma(n)$也满足差分方程

$$\sigma^2(n)=\alpha\sigma^2(n-1)+x^2(n-1) \tag{8.19}$$

为保证稳定性，要求 $0<\alpha<1$，参数 α 的取值影响 $\sigma(n)$的变化速度，例如，取 $\alpha=0.9$ 时，系统自适应的速度要比 $\alpha=0.99$ 时快得多，它们可分别对应于瞬时自适应和音节自适应。但是值得注意的是，$\sigma(n)$的变化快慢是由低通滤波器带宽所决定的，它又决定了 $\Delta(n)$和 $G(n)$所需的取样率。研究 $\Delta(n)$或 $G(n)$的最低取样率是重要的，因为 $\Delta(n)$或 $G(n)$必须作为边信息传送，它们将影响整个编码系统的数码率。如果 $\Delta(n)$是按帧估算的话(一般 10～30ms 为一帧)，则边信息所需的比特率就很低了。此外，为了在 40dB 信号动态范围内保持一个相对稳定的 SNR，那么要求 $\Delta(n)$或 $G(n)$的变化范围，即 $\Delta_{max}/\Delta_{min}$或 G_{max}/G_{min}值应达到 100。

(2) 反馈自适应

反馈型 PCM 系统如图 8.3 所示，其特点是输入信号的方差是由量化器输出或等效地由样本码序列估算出来的，如同前馈系统一样，量化间隔 $\Delta(n)$和增益 $G(n)$也按式(8.15)变化。这个方案的优点是：$\Delta(n)$或 $G(n)$无须保存或传送，因为编码端可以如同解码端那样直接从码序列中估算出 $\sigma^2(n)$来。由于不涉及数码率增加的问题，反馈自适应中的 $\Delta(n)$或 $G(n)$总是逐点自适应修正，以求得较好的自适应效果。反馈自适应方案的缺点是：对码序列中由于传输产

生的误差比较敏感，因为误码还将影响到 $\Delta(n)$或 $G(n)$的自适应，并且这一影响会不断地传播下去。

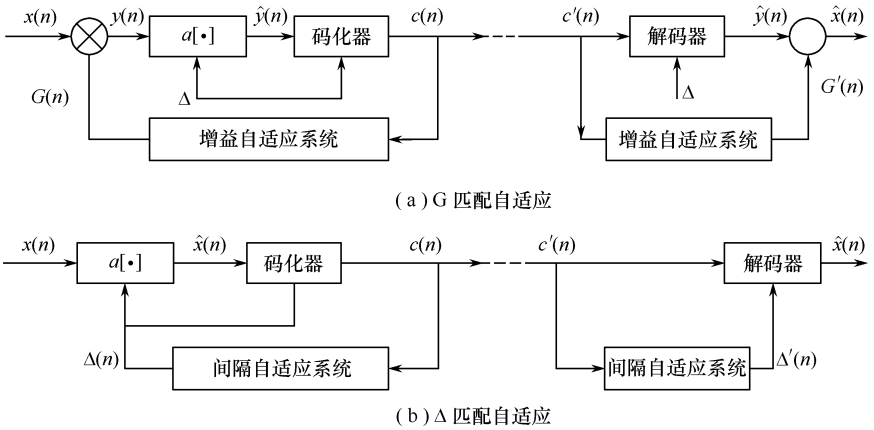


图 8.3　两种反馈自适应量化方框图

一般来讲，前馈自适应和反馈自适应相比，信噪比略高一些；但是前馈自适应需要延迟一段时间去计算短时方差，而反馈自适应则是瞬时完成的。总之，自适应量化能给出超过 μ-律或 A-律量化的信噪比，适当选定 $\Delta_{max}/\Delta_{min}$，也可使自适应动态范围与后者相当，选择较小的 Δ_{min}还可使无语言活动时量化噪声很低，因此自适应量化是一种很有效的编码方法。

8.4.2　自适应预测编码 APC

1. 基本的自适应预测编码系统

我们在讨论语音信号的线性预测分析原理时，假定一个语音样本 $s(n)$可以近似地被它过去的 p 个样本的线性组合所预测，预测样本值

$$\tilde{s}(n) = \sum_{i=1}^{p} a_i s(n-i) \tag{8.20}$$

式中，$a_i(1 \leqslant i \leqslant p)$称为预测系数，$p$ 是预测阶数，令 $e(n)$表示实际值与预测值之间的误差

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{8.21}$$

$e(n)$即线性预测误差，也被称为线性预测残差。对式(8.21)两边取变换后有

$$E(z) = \left[1 - \sum_{i=1}^{p} a_i z^{-i}\right] S(z) = A(z)S(z) \tag{8.22}$$

式中

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \tag{8.23}$$

因此，$e(n)$可以让语音信号 $s(n)$通过一个全零点的滤波器 $A(z)$而得到。可以设想，如式(8.20)预测效果很好的话，那么预测残差 $e(n)$的幅度变化范围和平均能量必定比原来的语音信号 $s(n)$要小；如果对残差序列 $e(n)$做量化和编码，在同样信号量化噪声比条件下，所需的量化比特数就可以减少，从而达到压缩编码的目的。基于这一原理的方法称为预测编码，当预测系数是自适应地随语音信号变化时，又称自适应预测编码。

自适应预测编码系统是如何提高信噪比的呢？我们用图 8.4 来说明。

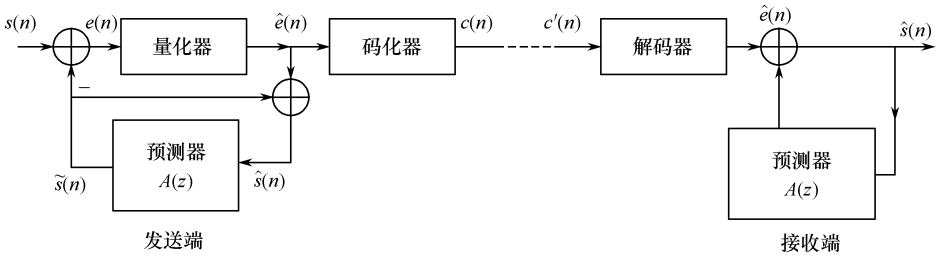


图 8.4　基本的自适应预测编码系统

从图 8.4 可以看到，不考虑传输信道的误码，系统解码后输出为

$$\begin{aligned}\hat{s}(n)&=\hat{e}(n)+\tilde{s}(n)=[e(n)+q(n)]+\tilde{s}(n)\\&=[s(n)-\tilde{s}(n)+q(n)]+\tilde{s}(n)=s(n)+q(n)\end{aligned}\tag{8.24}$$

式中，$q(n)$是残差信号 $e(n)$的量化误差

$$q(n)=\hat{e}(n)-e(n)\tag{8.25}$$

注意重构的信号$\hat{s}(n)$在编码端和解码端都可以得到。根据信号量化噪声比的定义有

$$\mathrm{SNR}=\frac{E[s^2(n)]}{E[q^2(n)]}=\frac{E[s^2(n)]E[e^2(n)]}{E[e^2(n)]E[q^2(n)]}=G_p\cdot SNR_q$$

$E[s^2(n)]$、$E[e^2(n)]$和 $E[q^2(n)]$分别是信号、残差和量化噪声的平均能量，不难看出，$\mathrm{SNR}_q=E[e^2(n)]/E[q^2(n)]$是量化器的信噪比，$G_p=E[s^2(n)]/E[e^2(n)]$是自适应预测增益。图 8.5 给出了固定预测和自适应预测两种情况下预测增益和预测阶数 p 的关系。

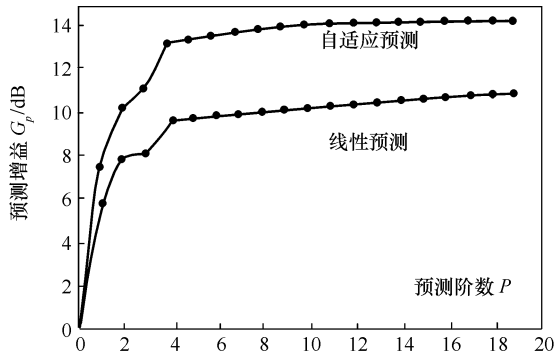


图 8.5　预测增益与预测阶数的关系

由图可见，阶数 $p>4$ 时，固定预测有 10dB 的增益，自适应预测有约 14dB 的增益。从以上分析可知，自适应预测编码有下列三个特性：

① 对同样比特数的量化器，APC 信噪比总是大于非预测编码，即 $G_p=E[s^2(n)]/E[e^2(n)]$总是大于 1。

②增益 G_p 是随时间变化的，因为它事实上是信号频谱的函数，谱的动态范围越大，信号样本之间相关性就越强，预测增益就越高。因此我们又把这种预测器称为基于频谱包络的预测。图 8.5 中 14dB 增益表示了整个讲话期间的最大值。

③量化噪声近似于白噪声，所以输出噪声的谱是平坦的。

2. 前馈与反馈自适应预测

与自适应量化器一样，自适应预测器也可分成前馈自适应和反馈自适应。前馈自适应预测器计算预测系数是通过误差

$$E=\sum_{n=0}^{N-1}e^2(n)=\sum_{n=0}^{N-1}\left[s(n)-\sum_{i=1}^{p}a_i s(n-i)\right]^2 \tag{8.26}$$

最小来求得。a_i 是按帧时变的，即按 10～30ms 为一帧来决定求和的样本点数 N 和系数。因为式(8.26)使用了输入语音信号 $s(n)$，它在接收端是得不到的，因此预测器系数必须作为边信息传输到接收端。对反馈自适应，预测器系数是从 $\hat{s}(n)$ 序列出发，使误差

$$\hat{E}=\sum_{n=0}^{N-1}\hat{e}^2(n)=\sum_{n=0}^{N-1}\left[\hat{s}(n)-\sum_{i=1}^{p}a_i\,\hat{s}(n-i)\right]^2 \tag{8.27}$$

最小求得。从图 8.4 看到，$\hat{s}(n)$ 在发送端与接收端都可以得到，因此除了传送 $\hat{e}(n)$，无须任何附加的边信息传给接收端。

为清楚起见，我们将前馈和反馈自适应预测方法做一下简单的比较。

① 前馈自适应预测的效果，一般讲略优于反馈自适应预测；但前馈预测的问题是必须传送预测系数到接收端。为了保证精确传送，就需适当地量化和编码它们，并和 $\hat{e}(n)$ 有效地组合起来，达到高效率的传输，这将使发送端变得比较复杂；而反馈预测则没有这个问题。

② $\hat{e}(n)$ 传输误码对反馈自适应预测编码的影响较大。在前馈自适应预测编码器中，$\hat{e}(n)$ 的误码不影响预测器系数。当然，预测器系数的传输本身也会出现误码；但它只局限于影响本帧的结果，而且一般说来，在编码预测器系数时都采取了有效措施，即使发生了误码也不至于造成系统的不稳定。反馈自适应预测算法求得的预测器系数，不能保证它们形成的合成滤波器一定是稳定的，同时要考虑算法的收敛性、有限字长的影响等，这都使得反馈自适应算法比较复杂。

8.4.3 自适应差分脉冲编码调制

1. 差分脉冲编码调制 DPCM

这是 APC 的一种特殊情况，它的预测器具有简单的形式：

$$A(z)=1-a_1z^{-1} \tag{8.28}$$

式中，a_1 是一个固定的常数，可以根据信号频谱的长期平均估算最优 $A(z)$ 而得到。在 DPCM 中，被量化和编码的是 $e(n)=x(n)-a_1x(n-1)$，即传送的是相邻样本的差值，所以又称为“差分脉冲编码调制”。因为 a_1 是固定的，显然它不可能对所有讲话者以及所有语音内容都是最佳的。采用高阶固定预测，改善效果并不明显；比较好的方法当然是采用高阶自适应预测。采用自适应量化及高阶自适应预测的 DPCM，又称为 ADPCM，它本质上也是自适应预测编码，即属于一种 APC 系统。

2. 增量调制 DM

增量调制基本上是一种 DPCM 方法，它与一般 DPCM 的主要区别有二点：一是增量调制中波形的取样率大大高于由取样定理确定的奈奎斯特取样速率，二是差值信号使用 2 个电平，亦即用 1 比特的量化器。由于取样率提高使得相邻样本之间的相关性变大，差值信号能量减小；从而允许只用 2 个电平去粗量化，实际上，DM 中传送的仅是差值信号的极性，即表征这个

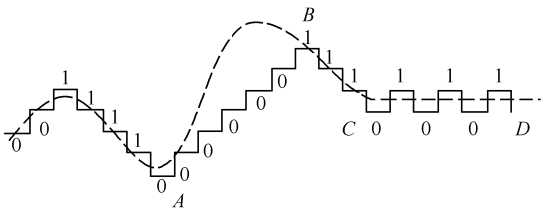


图 8.6　增量调制示意图

取样值比上一个取样值是增加了还是减少了；在接收端根据传输的极性符号，在前一个取样值上增加或减小一个增量即可。因此，DM 系统的比特率就等于波形的取样率，图 8.6给出了 DM 的编码情况。图 8.6 是一段原始语音信号（虚线）和根据增量调制编码序列所恢复的阶梯信号的波形，各阶梯的高度等于编码器中的量化电平 Δ。在均匀量化时，Δ 的大小与信号电平无关，始终保持恒定，因而 $x(n)$的量化值$\hat{x}(n)$构成的增加和减小都将是线性的。这样，在译码器中，所恢复的阶梯波的上升或下降有可能跟不上信号的变化，因而产生滞后，这就造成了失真，称为“斜率过载”失真，如图 8.6 的 AB 段。斜率过载期间的码字将是一连串的 0 或一连串的 1。为了避免这种失真，要求阶梯波的上升和下降的斜率等于或大于语音信号的最大变化斜率，即

$$\frac{\Delta}{T}\geqslant \max\left|\frac{\mathrm{d}x_{\mathrm{a}}(t)}{\mathrm{d}t}\right| \tag{8.29}$$

式中，$x_{\mathrm{a}}(t)$是原始模拟语音信号，T 是其取样时间间隔。

当语音信号不发生变化或变化很缓慢时，预测误差信号将等于零或具有很小的绝对值。这种情况下预测误差信号被量化为 Δ 和 $-\Delta$ 的概率是相等的，因此，经量化后成为幅度为 2Δ 的等幅振荡，编码为 0 和 1 交替出现的序列。在译码器中所得到的将是峰－峰值等于 2Δ 的等幅脉冲序列。这便形成一种噪声，称为“颗粒噪声”，如图 8.6 的 CD 段所示。

从式(8.29)看出，为减小斜率过载失真，要求选取较大的 Δ 值；而为减小颗粒噪声，却应当将 Δ 值取得小些。这是相互矛盾的。因此，通常需要对这两方面的要求折中加以考虑。

一般情况下，人的听觉器官不易察觉斜率过载失真，而颗粒噪声在整个音频范围内都会产生影响，对音质影响严重。因此，常常将 Δ 取得尽可能小（但应当与语音信号电平相匹配）。与此同时，也要兼顾到斜率过载失真不能太严重。在 Δ 选定后，如果斜率过载失真太严重，以至于无法接受，这时可以用加大取样频率的办法来降低斜率过载失真[因为从式(8.29)看出，T 的减小可以减小斜率过载失真]。然而，应当注意到不要因此让比特率增加得过多。

3. 自适应增量调制 ADM

ADM 的基本思想是：使增量 Δ 自适应语音信号的平均斜率变化，当信号波形平均斜率变大时，Δ 自动增大、反之则减小；从而缓解 DM 中由于 Δ 固定引起的矛盾。ADM 一般采用反馈自适应方式，即增量 Δ 由量化后的代码来控制，例如

$$\Delta(n)=M\Delta(n-1),\quad \text{其中 } \Delta(n)\text{满足 } \Delta_{\min}\leqslant\Delta(n)\leqslant\Delta_{\max} \tag{8.30}$$

这里 $\Delta_{\max}$，$\Delta_{\min}$是预先确定的增量的上下限，乘数 M 是当前码字 $c(n)$和前一个码字 $c(n-1)$的函数，一般选择

$$\text{若：}\begin{cases} c(n)=c(n-1)=c(n-2), & \text{则 } M>1 \\ c(n)\neq c(n-1), & \text{则 } M<1 \end{cases} \tag{8.31}$$

另一种自适应增量调制是所谓“连续可变斜率增量调制”(CVSD)，它的自适应规则是

$$\left.\begin{aligned} &\Delta(n)=\beta\Delta(n-1)+D_2, c(n)=c(n-1)=c(n-2) \\ &\Delta(n)=\beta\Delta(n-1)+D_1, \text{其他} \end{aligned}\right\} \tag{8.32}$$

这里，$0<\beta<1$，$D_2\gg D_1>0$；$\Delta(n)$递推公式中的最小值和最大值是固定的。与前面一样，其基本原理是：按照码序中表示斜率过载的情况增大增量，假定接连三个码字是“1”或者全是“0”，

则增量 $\Delta(n)$增加一个量，不出现这种码序时，$\Delta(n)$一直减小到 Δ_{min}（因为 $\beta<1$）。参数 β 控制自适应的速度，若 β 接近于 1，则 $\Delta(n)$的增加和衰减速率减慢；但若 β 比 1 小很多，则自适应速度加快。

CVSD 编码器在数码率低于 2.4kbit/s 时，产生的语音质量优于 APC 编码器，主要是颗粒噪声低，听起来比较清晰；但是在 16kbit/s 的数码率，CVSD 的语音质量要比相同数码率下的 APC 编码器差。

4. 自适应差分脉冲编码调制 ADPCM

在许多应用中，特别是长途传输系统，64kbit/s 的 G.711 标准占用的频带太宽，通信成本太贵。ITU-T 从 1981 年起经过三年的讨论与研究，于 1984 年提出了 G.721 32kbit/s ADPCM 编码标准，并于 1986 年根据两年间运行中出现的问题做了进一步修正。

ADPCM 将脉冲编码调制、差值调制和自适应技术三者结合起来，进一步利用语音信号样点间的相关性，并针对语音信号的非平稳特点，使用了自适应预测和自适应量化，在 32kbit/s 速率上能够给出网络等级语音质量，从而符合进入公用网的要求。图 8.7 是 G.721 算法的框图，其中虚线部分是解码器框图。由图中可以看出，编码器中嵌入一个解码器，使得编码器的自适应修正完全取决于信号的反馈值。这个反馈值与解码器的输出是一致的，所以后续的差值采样就补偿了量化误差，从而避免了量化误差的积累。

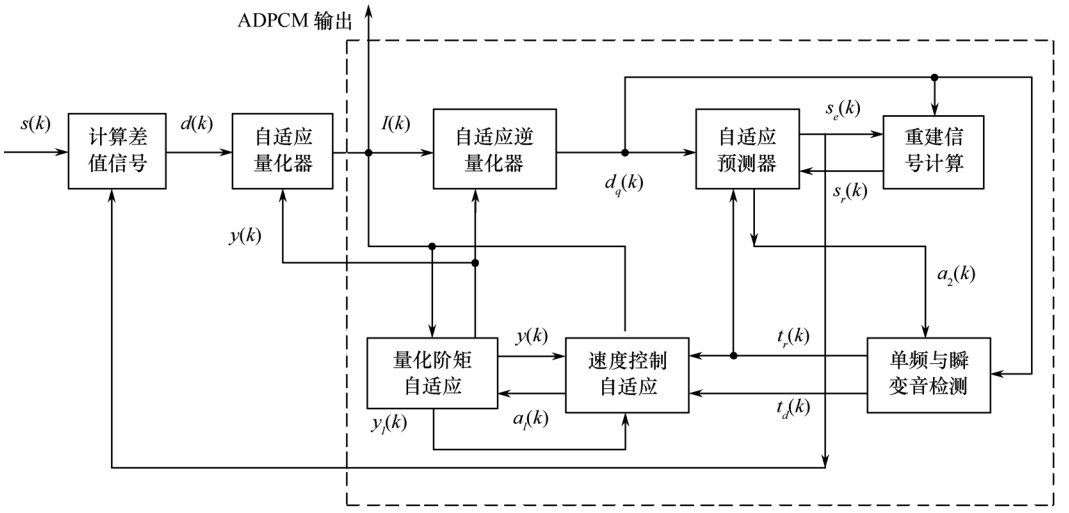


图 8.7　G.721 编码器原理框图

下面详细介绍 G.721 各部分算法。

① 求采样值 $s(k)$与其估值 $s_e(k)$之差

$$d(k)=s(k)-s_e(k) \tag{8.33}$$

② 自适应量化 $d(k)$并编码输出 $I(k)$

$$I(k)=\log_2|d(k)|-y(k) \tag{8.34}$$

其中，$I(k)$还含有一位符号。表 8.2 给出 $I(k)$的编码值。$y(k)$是量化阶矩自适应因子，它由调整短时能量变化较快的语音信号的 $y_u(k)$和调整数据类慢变信号的 $y_l(k)$两部分，经速度调整因子 $a_l(k)$加权平均而成：

$$y(k)=a_l(k)\cdot y_u(k-1)+[1-a_l(k)]y_l(k-1)\quad 0\leqslant a_l\leqslant 1 \tag{8.35}$$

对快变信号，$a_l(k)$趋于1，而对慢变信号 $a_l(k)$趋于0。

表 8.2　G.721 编码器量化表

归一化输入 $\log_2\|d(k)\|-y(k)$	输出代码 $I(k)$	归一化量化输出 $\log_2\|d_q(k)\|-y(k)$
[3.12，+∞]	7	3.32
[2.72，3.12]	6	2.91
[2.34，2.72]	5	2.52
[1.91，2.34]	4	2.13
[1.38，1.91]	3	1.66
[0.62，1.38]	2	1.05
[−0.98，0.62]	1	0.031
[−∞，−0.98]	0	−∞

③ 阶矩自适应因子

$y_u(k)$称快速非锁定标度因子，它的取值范围在 $1.06\leqslant y_u(k)\leqslant 10$ 区间，对应的线性域为 $\Delta\min=2^{1.06}=2.085$，$\Delta\max=2^{10}=1024$。

$$y_u(k)=(1-2^{-5})y(k)+2^{-5}w[I(k)] \tag{8.36}$$

$w[I(k)]$的取值如表 8.3 所示。

表 8.3　$w[I(k)]$的取值

$\|I(k)\|$	7	6	5	4	3	2	1	0
$w[I(k)]$	70.13	22.19	12.38	7.00	4.00	2.56	1.13	−0.75

为了适应语音预测差值信号中的基音引起的能量突变，$w[I(k)]$的高端取值都很大。对于带内数据，信号短时能量基本上是平稳的，阶矩自适应采用如下算法：

$$y_l(k)=(1-2^{-6})y_l(k-1)+2^{-6}y_u(k) \tag{8.37}$$

式中，$y_l(k)$称为锁定标度因子。

④ 速度控制

$a_l(k)$是速度控制因子，它是通过 $I(n)$的长时平均幅度值 $d_{\mathrm{ml}}(k)$与短时平均幅度值$d_{\mathrm{ms}}(k)$的差求出的。它反映了预测余量信号的变化率。

长时：
$$d_{\mathrm{ml}}(k)=(1-2^{-7})d_{\mathrm{ml}}(k-1)+2^{-7}F[I(k)] \tag{8.38}$$

短时：
$$d_{\mathrm{ms}}(k)=(1-2^{-5})d_{\mathrm{ms}}(k-1)+2^{-5}F[I(k)] \tag{8.39}$$

函数 $F[I(k)]$的取值如表 8.4 所示。

表 8.4　$F[I(k)]$的取值

$\|I(k)\|$	7	6	5	4	3	2	1	0
$F[I(k)]$	7	3	1	1	1	0	0	0

当余量信号短时能量平稳时，$I(k)$的统计特性随时间变化很小，$d_{\mathrm{ml}}(k)$与 $d_{\mathrm{ms}}(k)$相差不大。而当余量信号短时能量起伏较大时，它们出现差值。利用这一特性先计算中间参数$a_p(k)$：

$$a_p(k)=\begin{cases}(1-2^{-4})a_p(k-1)+2^{-3}, & \text{当}|d_{ms}(k)-d_{ml}(k)|\geqslant 2^{-3}d_{ml}(k)\\ & \text{或当 } y(k)<3\\ (1-2^{-4})a_p(k-1), & \text{其他情况}\end{cases} \tag{8.40}$$

显然，当 $I(k)$ 幅度变化较大时 $a_p(k)\to 2$，而差别较小时 $a_p(k)\to 0$。条件 $y(k)<3$ 表明输入信号很小，处于清音段或噪声段，这时也有 $a_p(k)\to 2$，以便使量化器处于快速自适应状态来等待输入信号的突然变化。量化器速度控制因子 $a_l(k)$ 是通过对 $a_p(k)$ 限幅得到：

$$a_l(k)=\begin{cases}1, & \text{当 } a_p(k-1)\geqslant 1\\ a_p(k-1), & \text{当 } a_p(k-1)<1\end{cases} \tag{8.41}$$

这样，量化器从快速自适应向慢速自适应转变有一个延迟。对于带内调幅数据，这种延迟效应可以防止自适应速度过早变慢，从而避免脉冲沿产生太大的畸变。

⑤ 自适应逆量化器输出

$$d_q(k)=2^{y(k)+I(k)} \tag{8.42}$$

⑥ 自适应预测

预测器采用 6 阶零点，二阶极点的模型。预测信号为

$$\begin{aligned}s_e(n) &= \sum_{i=1}^{2}a_i(n-1)s_r(n-i)+s_{ez}(n)\\ s_{ez}(n) &= \sum_{j=1}^{6}b_j(n-1)d_q(n-j)\end{aligned} \tag{8.43}$$

重建信号为

$$s_r(n)=s_e(n)+d_q(n) \tag{8.44}$$

极点，零点预测器系数分别是 a_i 和 b_j。其调整方式为

$$b_j(n)=(1-2^{-8})b_j(n-1)+2^{-7}\text{sgn}[dq(n)]\cdot\text{sgn}[dq(n-j)] \tag{8.45}$$

此式隐含差 $|b_j(n)|\leqslant 2$，为保证算法稳定，二阶极点预测器系数限制如下

$$|a_2(n)|\leqslant 0.75;\quad |a_1(n)|\leqslant 1-a_2(n)-2^{-4}$$

它们的调整方式为

$$a_1(n)=(1-2^{-8})a_1(n-1)+3\cdot 2^{-8}\text{sgn}[p(n)]\cdot\text{sgn}[p(n-1)] \tag{8.46}$$

$$\begin{aligned}a_2(n)=&(1-2^{-7})a_2(n-1)+2^{-7}\text{sgn}[p(n)]\cdot\\ &\{\text{sgn}[p(n-2)]-f[a_1(n-1)]\cdot\text{sgn}[p(n-1)]\}\end{aligned} \tag{8.47}$$

式中

$$p(n)=dq(n)+s_{ez}(n) \tag{8.48}$$

$$f(a_1)=\begin{cases}4a_1, & \text{当}|a_1|\leqslant\dfrac{1}{2}\\ 2\text{sgn}[a_1], & \text{当}|a_1|>\dfrac{1}{2}\end{cases} \tag{8.49}$$

⑦ 单频和瞬变调整

当 ADPCM 编码器遇到频移键控信号(FSK)或其他窄带瞬变信号时，需要将系统从慢速自适应状态强制性地调整到快速自适应状态。为此，引入单频信号判定条件 t_d 和窄带信号瞬变判据 t_r：

$$t_d(n)=\begin{cases}1, \text{若 } a_2(n)<-0.71875\\ 0, \text{其他}\end{cases} \tag{8.50}$$

$$t_r(n)=\begin{cases}1, t_d(n)=1 \text{ 同时} |d_q(n)|>24.2^{y_l(n)}\\ 0, \text{其他}\end{cases} \tag{8.51}$$

当 $t_d(n)=1$ 时，认为出现了单频信号或频率瞬变。这时强制将量化器处于快速自适应状态。当 $t_r(n)=1$ 时，还需将 $a_i(n)$ 和 $b_j(n)$ 同时置零。采用这些措施后，G.721 ADPCM 可以传递 4.8kbit/s 的 FSK 信号。同时 a_p 的判定也由下式决定：

$$a_p(n)=\begin{cases}(1-2^{-4})a_p(n-1)+2^{-3} & \text{；若} |d_{ms}(n)-d_{ml}(n)|\geqslant 2^{-3}d_{ml}(n) \\ & \text{或 } y(n)<3 \text{ 或 } t_d(n)=1 \\ 1 & \text{；} \quad t_r(n)=1 \\ (1-2^{-4})a_p(n-1) & \text{；} \quad \text{其他}\end{cases} \tag{8.52}$$

当 ADPCM 与 PCM 之间发生换码级联时，需要在 ADPCM 内部进行 PCM 级联同步调整。方法是在解码端将重建信号 $s_r(n)$ 重新编码成 ADPCM 码 $I_{dx}(n)$ 并与输入的 $I(n)$ 比较，根据差值调整重建信号 $s_r(n)$ 的电平级别。经过同步调整过程，ADPCM 可以有效地防止同步级联误差累积。

5. G.721 ADPCM 语音编码标准的 MATLAB 实现

为了便于理解 G.721 的 MATLAB 程序，特对各模块程序功能介绍如下：

d.m 主函数程序文件，完成赋初值、信号输入及调用语音编解码函数，在 MATLAB 中加载 G.721MATLAB 程序文件后，在命令窗口中输入 d 并回车，即可完成 G.721 语音编解码算法。

adpcm.m 语音编解码函数文件

Sek_com.m 自适应预测

Dk_com.m 采样值与其估值差值计算

yu_result.m 快速非锁定标度因子计算

yl_result.m 锁定标度因子计算

Tdk_com.m 单频信号判定

Trk_com.m 窄带信号瞬变判定

Alk_com.m 自适应速度控制与自适应预测

Yk_com.m 量化阶矩自适应因子计算

Ik_com.m 自适应量化并编码输出

Dqk_com.m 自适应逆量化器输出

Srk_com.m 重建信号输出

f_com.m 自适应预测中 f 函数值计算

sgn_com.m 算法中用到的符号函数

wi_result.m 量化器标度因子自适应 wi 的选取

fi_ result.m 速度控制中 F[I(k)]计算

【程序 8.1】 主函数程序 d.m

```
clc
clear
coe=[1,0,1,0,0,0,0,0,0,0,0];                    % 初始化系数
coe1=[0,0,0] ;
coe2=[0,0,0,0,0,0,0,0,0,0,0];
```

```
coe3=[0];
Dqk=zeros(1,7);
fid=fopen('zhongguo.txt','rt');            % 读文件,文件格式为.txt
a=fscanf(fid,'%e\n');
fclose(fid);
% fid=('ling11.wav');wavwrite(44100,fid);    % 转换回 wav 格式音频文件
fid=fopen('zhongguo.721.txt','wt');
for  i=1:size(a,1)
    Slk=a(i);          % 输入信号
    [coe,coe1,coe2,coe3,Dqk]=adpcm(Slk,coe,coe1,coe2,coe3,Dqk);
    % 调用语音编解码函数
    fprintf(fid,'%f\n',coe2(5));
end
fclose(fid)
% ————————————————波形显示————————————————
fid=fopen('zhongguo.txt','rt');
a=fscanf(fid,'%e\n');
fid=fopen('zhongguo.721.txt','rt');
b=fscanf(fid,'%e\n');
subplot(211),plot(a);
title('输入语音波形');
subplot(212),plot(b);
title('解码输出波形');
```

语音编解码子函数程序 adpcm.m

```
function [coe,coe1,coe2,coe3,Dqk]=adpcm(Slk,coe,coe1,coe2,coe3,Dqk)
                                                          % 语音编解码函数
Yk_pre=coe2(1);    % 初值传递
Sek_pre=coe2(2);
Ik_pre=coe2(3);
Ylk_pre_pre=coe2(4);
Srk_pre=coe2(5);
Srk_pre_pre=coe2(6);
a2=coe2(7);
Tdk_pre =coe2(8);
Trk_pre =coe2(9);
Num=coe2(10);

        coe2(10)=coe2(10)+1;
        [Sek,coe]=Sek_com(Srk_pre,Srk_pre_pre,Dqk,coe);  % 自适应预测

        Dk=Dk_com( Slk, Sek );   % 采样值与其估值差值计算

        Yuk_pre=yu_result( Yk_pre, wi_result(abs(Ik_pre)) );   % 快速非锁定标度
                                                               % 因子计算
```

```
if   Yuk_pre< 1.06
     Yuk_pre=1.06;
elseif  Yuk_pre> 10.00
    Yuk_pre=10.00;
end

Ylk_pre=yl_result( Ylk_pre_pre, Yuk_pre );     % 锁定标度因子计算
Trk_pre=Trk_com( a2, Dqk(6), Ylk_pre );  % 窄带信号瞬变判定
Tdk_pre=Tdk_com( a2 );     % 单频信号判定
[Alk,coe1]= Alk_com( Ik_pre, Yk_pre ,coe1,Tdk_pre,Trk_pre);
% 自适应速度控制与自适应预测

if  Alk< 0.0
    Alk=0.0;
elseif  Alk> 1.0
    Alk=1.0;
end

[Yk,coe3]=Yk_com(Ik_pre,Alk,Yk_pre,coe3);      % 量化阶矩自适应因子计算

Ik=Ik_com( Dk, Yk );      % 自适应量化并编码输出

Yk_pre=Yk;
Srk_pre_pre=Srk_pre;
Sek_pre=Sek;
Ylk_pre_pre=Ylk_pre;
Ik_pre=Ik;

coe2(1)= Yk;
coe2(6)= Srk_pre;
coe2(2)= Sek;
coe2(4)= Ylk_pre;
coe2(3)= Ik;

Dqk(1)=Dqk(2);
Dqk(2)=Dqk(3);
Dqk(3)=Dqk(4);
Dqk(4)=Dqk(5);
Dqk(5)=Dqk(6);
Dqk(6)=Dqk(7);

Dqk(7)=Dqk_com( Ik_pre,Yk_pre);      % 自适应逆量化器输出
Srk_pre=Srk_com( Dqk(7), Sek_pre);     % 重建信号输出
coe2(5)=Srk_pre;
```

自适应预测子函数程序 Sek_com. m

```
function[g,f]=Sek_com(Srk_pre,Srk_pre_pre,Dqk,coe)
% 自适应预测函数
  a1_pre=coe(1);
  a2_pre=coe(2);
  b1_pre=coe(3);
  b2_pre=coe(4);
  b3_pre=coe(5);
  b4_pre=coe(6);
  b5_pre=coe(7);
  b6_pre=coe(8);
  Sezk_pre=coe(9);
  p_pre2 =coe(10);
  p_pre3=coe(11);
% 6 阶零点预测器系数
  b1=(1 - 2^(-8)) * b1_pre+2^(-7) * sgn_com(Dqk(7)) * sgn_com(Dqk(6));
  b2=(1 - 2^(-8)) * b2_pre+2^(-7) * sgn_com(Dqk(7)) * sgn_com(Dqk(5));
  b3=(1 - 2^(-8)) * b3_pre+2^(-7) * sgn_com(Dqk(7)) * sgn_com(Dqk(4));
  b4=(1 - 2^(-8)) * b4_pre+2^(-7) * sgn_com(Dqk(7)) * sgn_com(Dqk(3));
  b5=(1 - 2^(-8)) * b5_pre+2^(-7) * sgn_com(Dqk(7)) * sgn_com(Dqk(2));
  b6=(1 - 2^(-8)) * b6_pre+2^(-7) * sgn_com(Dqk(7)) * sgn_com(Dqk(1));

% 2 阶极点预测器系数
  Sezk=b1 * Dqk(7)+b2 * Dqk(6)+b3 * Dqk(5)+b4 * Dqk(4)+b5 * Dqk(3)+b6 * Dqk(2);
  p_pre1=Dqk(7)+Sezk_pre;
  if abs(p_pre1)< =0.000001
      a1=(1 -2^(-8)) * a1_pre;
      a2=(1 -2^(-7)) * a2_pre;
  else
      a1=(1 -2^(-8)) * a1_pre+(3 * 2^(-8)) * sgn_com(p_pre1) * sgn_com(p_pre2);
      a2=(1 - 2^(-7)) * a2_pre+2^(-7) * (sgn_com(p_pre1) * sgn_com(p_pre3)
          - f_com(a1_pre) * sgn_com(p_pre1) * sgn_com(p_pre2));
  end
% 自适应预测和重建信号计算器
  coe(1)= a1;
  coe(2)= a2;
  coe(3)= b1;
  coe(4)= b2;
  coe(5)= b3;
  coe(6)= b4;
  coe(7)= b5;
  coe(8)= b6;
  coe(9)= Sezk;
  coe(10)=p_pre1;
```

```
    coe(11)=p_pre2;
    g=(a1*Srk_pre+a2*Srk_pre_pre+Sezk);
    f=coe;
```

采样值与其估值差值计算子函数 Dk_com.m

```
function d=Dk_com(Slk,Sek) % 采样值与其估值差值计算函数
Dk=Slk-Sek;
d=Dk;
```

快速非锁定标度因子计算子函数 yu_result.m

```
function yu=yu_result(y_now,wi_now)      % 快速非锁定标度因子计算函数
yu=(1-2^(-5))*y_now+2^(-5)*wi_now;
yu=yu;
```

锁定标度因子计算子函数 yl_ result.m

```
function yl=yl_result(yl_pre,yu_now)      % 锁定标度因子计算函数
yl=(1-2^(-6))*yl_pre+2^(-6)*yu_now;
yl=yl;
```

单频信号判定子函数 Tdk_com.m

```
function Tdk=Tdk_com(A2k)       % 单频信号判定函数
    if (A2k<-0.71875) Tdk=1;
    else Tdk=0;
    end
    Tdk=Tdk;
```

窄带信号瞬变判定子函数 Trk_com.m

```
function Trk=Trk_com(A2k,Dqk,Ylk)     % 窄带信号瞬变判定
    if ((A2k<-0.71875)&(fabs(Dqk)> pow(24.2,Ylk)))   Trk=1;
    else     Trk=0;
    end
    Trk=Trk;
```

自适应速度控制与自适应预测子函数 Alk_com.m

```
function [h,coe1]=Alk_com(Ik_pre,Yk_pre,coe1,Tdk_pre,Trk_pre)  % 量化器速度控
                                                                % 制函数
    Dmsk_p2=coe1(1);
    Dmlk_p2=coe1(2);
    Apk_pre2=coe1(3);
    Dmsk_p1=(1-2^(-5))*Dmsk_p2+2^(-5)*fi_result(abs(Ik_pre));
                                                        % Ik短时平均幅度值
Dmlk_p1=(1-2^(-7))*Dmlk_p2+2^(-7)*fi_result(abs(Ik_pre));
                                                        % Ik长时平均幅度值
coe1(1)= Dmsk_p1;
coe1(2)=Dmlk_p1;

    if ((abs(Dmsk_p1-Dmlk_p1)>=2^(-3)*Dmlk_p1)|(Yk_pre<3)|(Tdk_pre==1))
        Apk_pre1=(1-2^(-4))*Apk_pre2+2^(-3);
     elseif  (Trk_pre == 1)   Apk_pre1=1;
```

```
            else Apk_pre1=(1 - 2^(-4)) * Apk_pre2;
    end
    coe1(3)= Apk_pre1;
    if  Apk_pre1> =1
        Alk=1;
    else  Alk=Apk_pre1;
    end
    h=Alk;
```

量化阶矩自适应因子计算子函数 Yk_com. m

```
function [Yk,coe3]=Yk_com(Ik_pre,Alk,Yk_pre,coe3)      % 量化阶矩自适应因子计算
Yl_pre_pre=coe3;
Yu_pre=(1 - 2^(-5)) * Yk_pre+2^(-5) * wi_result(abs(Ik_pre));
                                                        % 快速非锁定标度因子计算
Yl_pre=yl_result(Yl_pre_pre,Yu_pre);                    % 锁定标度因子计算
coe3=Yl_pre;
Yk=Alk * Yu_pre+(1 - Alk) * Yl_pre;
```

自适应量化并编码输出子函数 Ik_com. m

```
function f=Ik_com(Dk, Yk)     % 编码输出函数
    if  Dk> 0   Dsk=0;
else    Dsk=1;
end
if Dk==0   Dk=Dk+0.0001;
end

Dlk=log(abs(Dk)) / log(2);
Dlnk=Dlk - Yk;          % 归一化输入
x=Dlnk;
a=10;
if  Dlnk<-0.98   Ik=0;          % 编码输出 Ik
end
if -0.98 <= Dlnk & Dlnk <   0.62    Ik=1;
end
if 0.62 <= Dlnk & Dlnk <   1.38     Ik=2;
end
if 1.38 <= Dlnk & Dlnk <   1.91     Ik=3;
end
if 1.91 <= Dlnk & Dlnk <   2.34     Ik=4;
end
if   2.34 <= Dlnk & Dlnk <   2.72    Ik=5;
end
if   2.72 <= Dlnk &Dlnk <   3.12     Ik=6;
end
if  Dlnk >= 3.12      Ik=7;
end
```

```
if Dsk == 1    Ik=-Ik;
end
f= Ik;
```

自适应逆量化器输出子函数 Dqk_com. m

```
function f=Dqk_com(Ik,Yk)  % 自适应逆量化器输出函数
if  Ik>=0     Dqsk=0;
       i=Ik;
else
       Dqsk=1;
       i=-Ik;
end
switch  i
case 7
         Dqlnk=3.32;
case 6
         Dqlnk=2.91;
case 5
         Dqlnk=2.52;
case 4
         Dqlnk=2.13;
case 3
         Dqlnk=1.66;
case 2
         Dqlnk=1.05;
case 1
         Dqlnk=0.031;
case 0
         Dqlnk=-1000;
     end
% 归一化量化输出
    Dqlk=Dqlnk+Yk;
    Dqk=2^Dqlk;
    if Dqsk==1
        Dqk=-Dqk;
    end
f=Dqk;
```

重建信号输出子函数 Srk_com. m

```
function Srk=Srk_com(Dqk,Sek)      % 重建信号计算函数
Srk=Dqk+Sek;
```

自适应预测中 f 函数值计算子函数 f_com. m

```
function b=f_com(a)          % f 函数值计算
if abs(a)<=0.5
     b=4*a;
else b=2*sgn_com(a);
```

```
    end
```

算法中用到的符号函数子函数 sgn_com. m

```
function b=sgn_com(a)        % 符号函数
if a> =0.000001     b=1;
else   b=-1;
end
```

量化器标度因子自适应 wi 的选取子函数 wi_result. m

```
function J=wi_result(in)
switch in
case 0
  wi=-0.75;
case 1
  wi=1.13;
case 2
  wi=2.56;
case 3
  wi=4.00;
case 4
  wi=7.00;
case 5
  wi=12.38;
case 6
  wi=22.19;
case 7
  wi=70.13;
end
J=wi;
```

速度控制中 F[I(k)]计算子函数 fi_ result. m

```
function w=fi_result(in)          % F[I(k)]计算函数
switch in

case 0
    fi=0;
case 1
    fi=0;
case 2
    fi=0;
case 3
    fi=1;
case 4
    fi=1;
case 5
    fi=1;
case 6
```

```
    fi=3;
case 7
    fi=7;
end
w=fi;
```

8.5 语音信号参数编码

基于参数编码理论的编码器由于其数码率比较低，通常称为声码器。最早的声码器是通道声码器，它是基于短时傅里叶变换的语音分析合成系统，由于其性能较差，现在已很少用。

根据语音信号的共振峰模型提出了共振峰声码器，该声码器通过对语音信号整体进行分析，提取共振峰的位置、幅度、带宽等参数，构成浊音和清音两个声道滤波器。浊音滤波器采用全极点滤波器，由多个二阶滤波器级联而成；清音滤波器一般采用一个极点和一个零点的数字滤波器。这些滤波器的参数都是时变的。与通道声码器相比，共振峰声码器合成出的语音质量更好，比特率更低。

在声码器中最具有代表性的是线性预测(LPC)声码器及其改进型。

8.5.1 LPC 声码器原理

LPC 声码器是应用最成功的低速率语音编码器。它基于全极点声道模型的假定，采用线性预测分析合成原理，对模型参数和激励参数进行编码传输。LPC 声码器遵循二元激励的假设，即浊音语音段采用间隔为基音周期的脉冲序列作为激励，清音语音段采用白噪声序列作为激励。因此，声码器只需对 LPC 参数、基音周期、增益和清浊音信息进行编码。LPC 声码器可以得到很低的比特率(2.4kbit/s 以下)。其工作原理如图 8.8 所示。

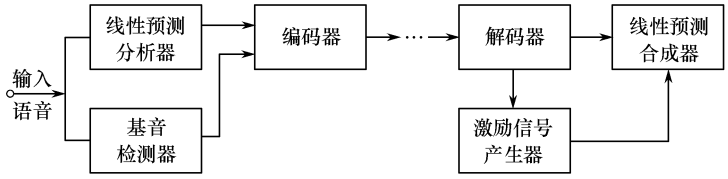


图 8.8 LPC 声码器原理图

虽然 LPC 声码器与 ADPCM 一样，都是基于线性预测分析来实现对语音信号的编码压缩，但是它们之间有着本质的区别，LPC 声码器不考虑重建信号波形是否与原来信号的波形相同，而努力使重建信号具有尽可能高的可懂度和清晰度，所以不必量化和传输预测残差，只需要传输 LPC 参数和重构激励信号的基音周期和清浊音信息。

LPC 声码器中，必须传输的参数是 p 个预测器系数、基音周期、清浊音信息和增益参数。直接对预测系数量化后再传输是不合适的，因为它的谱灵敏度极不均匀，有些系数很小的变化，就可能会引起频谱发生很大的变化。而且线性预测系数的内插特性也很差，内插得到的新参数，不一定能够构成稳定的合成滤波器。为此，可将预测器系数变换成其他更适合于编码和传输的参数形式，可参见第 6 章的内容。

8.5.2 LPC-10 编码器

LPC 声码器在通信领域，尤其是军事通信领域得到了广泛的应用。1976 年美国确定用 LPC 声码器标准 LPC-10 作为 2.4kbit/s 速率上的推荐编码方式。1981 年这个算法被官方接受，作为联邦政府标准 FS-1015 被公布。利用这个算法可以合成清晰、可懂的语音，但是抗噪声能力和自然度比较差。自 1986 年以来，美国第三代保密电话装置采用了速率为 2.4kbit/s 的 LPC-10e(LPC-10 的增强型)作为语音处理手段。下面介绍图 8.9 所示的 LPC-10 的工作原理和一些改进措施。

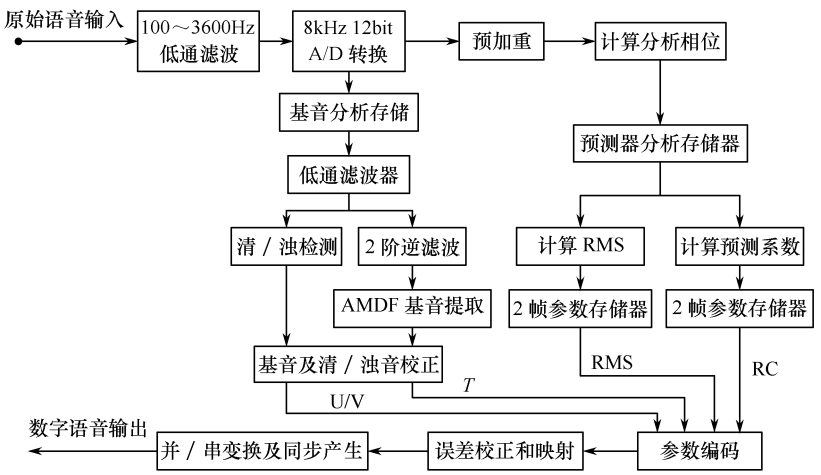


图 8.9 LPC-10 的编码器框图

1. 编码器

(1) 编码器基本原理

图 8.9 为 LPC-10 的编码器框图。原始语音经过 100～3600Hz 的锐截止的低通滤波器之后，输入 A/D 转换器，以 8kHz 采样率 12bit 量化得到数字化语音，然后每 180 个采样点(22.5ms)为一帧，以帧为处理单元。编码器分两个支路同时进行，其中一个支路用于提取基音周期 T 和清浊音 U/V 判决信息；另一支路用于提取声道滤波器参数 RC 和增益因子 RMS。提取基音周期的支路把 A/D 变换后输出的数字化语音缓存，经过低通滤波、二阶逆滤波后，再用平均幅度差函数 AMDF 计算基音周期，经过平滑、校正得到该帧的基音周期。与此同时，对低通滤波后输出的数字语音进行清浊音标志。提取声道参数支路需先进行预加重处理。预加重的目的是加强语音谱中的高频共振峰，使语音短时谱以及 LPC 分析中的残差频谱变得更为平坦，从而提高了谱参数估值的精确性。预加重滤波器的传递函数为

$$H_{p\omega}(z)=1-0.9375z^{-1} \tag{8.53}$$

(2) 计算声道滤波器参数

采用 10 阶 LPC 分析滤波器，利用协方差法对 LPC 分析滤波器 $A(z)=1-\sum_{i=1}^{10}a_iz^{-i}$ 计算预测系数 $a_1,a_2,\cdots,a_{10}$，并将其转换成反射系数 RC，或者部分相关系数 PARCOR 来代替预测系数进行量化编码。理论上 RC 参数和 PARCOR 参数互为相反数，系统稳定条件是其绝对

值小于1，这在量化时是容易保证的。LPC分析采用半基音同步算法，即浊音帧的分析帧长取为130个样本以内的基音周期整数倍值，来计算RC和RMS。这样，每一个基音周期都可以单独用一组系数处理。在接收端恢复语音时也是如此处理。清音帧是取长度为22.5ms的整帧中点为中心的130个样本形成分析帧来计算RC和RMS。

(3) 增益因子RMS的计算

用如下公式计算RMS：

$$\mathrm{RMS}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}x^2(i)} \tag{8.54}$$

式中，$x(i)$是经过预加重的数字语音；N是分析帧的长度。

(4) 基音周期提取和清/浊音检测

输入数字语音经3dB截止频率为800Hz的4阶Butterworth低通滤波器滤波，滤波后的信号再经过二阶逆滤波(逆滤波器的系数为前面LPC分析得到的短时谱参数$a_1,a_2,\cdots,a_{10}$)。把取样频率降低至原来的1/4，再计算延迟时间为20～256个样点的平均幅度差函数AMDF，由AMDF的最小值确定基音周期。计算AMDF的公式为

$$\mathrm{AMDF}(k)=\sum_{m=1}^{130}\left|x(m)-x(m+k)\right| \tag{8.55}$$

式中，$\tau=20,21,22,\cdots,40,42,44,\cdots,80,84,88,\cdots,156$。这相当于在50～400Hz范围内计算60个AMDF值。清/浊音判决是利用模式匹配技术，基于低带能量、AMDF函数最大值与最小值之比、过零率做出的。最后对基音值、清/浊音判决结果用动态规划算法，在3帧范围内进行平滑和错误校正，从而给出当前帧的基音周期T、清/浊音判决参数U/V。每帧清/浊音判决结果用两位码表示4种状态，这4种状态为：00——稳定的清音；01——清音向浊音转换；10——浊音向清音转换；11——稳定的浊音。

表8.5 LPC-10的比特数分配/bit

	清音	浊音
T/Voicing	7	7
RMS	5	5
Sync	1	1
k_1	5	5
k_2	5	5
k_3	5	5
k_4	5	5
k_5	4	
k_6	4	
k_7	4	
k_8	4	
k_9	3	
k_{10}	2	
误差校正	0	20
总计	54	53

(5) 参数编码与解码

在LPC-10的传输数据流中，将10个反射系数($k_1,k_2,\cdots,k_{10}$)、增益因子(RMS)、基音周期T、清/浊音U/V、同步信号Sync编码成每帧54bit。由于传输速率为44.4帧/s，因此，码率为2.4kbit/s。同步信号采用相邻帧1,0码交替的模式。表8.5是浊音帧和清音帧的比特数分配。

2. 解码器

LPC-10接收端解码器框图如图8.10所示。接收到的语音信号经串/并变换及同步后，利用查表法对数码流进行检错、纠错。纠错译码后的数据经参数解码得到基音周期、清/浊音标志、增益以及反射系数的数值，解码结果延时一帧输出。输出数据在过去的一帧、当前帧和将来的一帧共3帧内进行平滑。由于每帧语音只传输一组参数，但一帧之内可能有不止一个基音周期，因此要对接收数值进行由帧块到基音块的转换和插值。

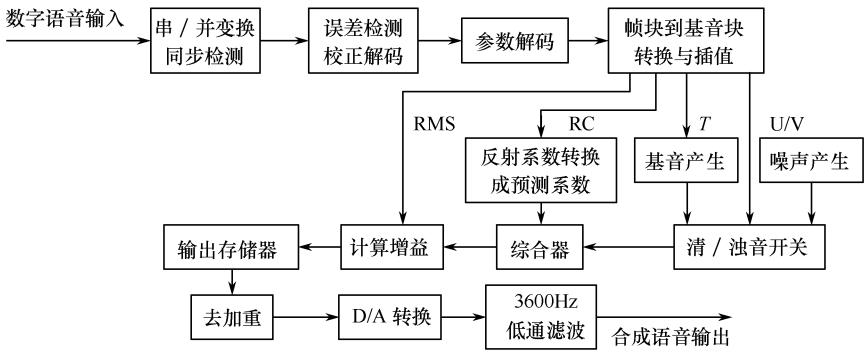


图 8.10　LPC-10 解码器框图

(1) 参数插值原则

对数面积比参数值每帧插值两次;RMS 参数值在对数域进行基音同步插值;基音参数值用基音同步的线性插值;在浊音向清音过渡时对数面积比不插值。每个基音周期更新一次预测系数、增益、基音周期、清/浊音等参数,这个过程在帧块到基音块的转换和插值中完成。

(2) 激励源

根据基音周期和清/浊音标志决定要采用的激励信号源。清音帧用随机数作为激励源;浊音帧用周期性冲激序列通过一个全通滤波器来生成激励源,这个措施改善了合成语音的尖峰性质。语音合成滤波器输入激励的幅度保持恒定不变,输出幅度受 RMS 参数加权。下面给出一组有 41 个样点的浊音激励信号:

$$\begin{aligned} e(n)=\{&0,0,0,0,0,0,0,0,5,-8,13,-24,43,-83,147,-252,359,\\ &-364,92,336,-306,-336,92,364,359,252,\\ &147,81,43,24,13,8,5,0,0,0,0,0,0,0,0\} \end{aligned}$$

若当前的基音周期不等于 41 个样点,则将此激励源截短或者填零,使之与基音周期等长。

(3) 语音合成

用 Levinson 递推算法将反射参数 $k_1,k_2,\cdots,k_p$,变换成预测系数 $a_1,a_2,\cdots,a_p$。接收端合成器应用直接型递归滤波器 $H(z)=1/(1-\sum_{i=1}^{p}a_iz^{-i})$ 合成语音。对其输出进行幅度校正、去加重,并变换为模拟信号,最后经 3600Hz 的低通滤波器后输出模拟语音。

3. LPC-10 编解码器的缺点及改进

LPC-10 虽然有编码速率低的优点,但是合成语音听起来很不自然,即使提高编码速率也无济于事。这主要是因为清浊音判决和浊音信号的基音检测很难做到十分可靠。有些摩擦音本身就清浊难分,在辅音与元音的过渡段或者有背景噪声的情况下,检测结果就更容易发生错误。这种错误对合成语音的清晰度影响特别严重。此外采用简单的二元激励形式,也不符合实际情况,因而造成自然度的下降。在增强型 LPC-10e 中采用了如下一些措施来改善语音的质量。

(1) 改善激励源

采用混合激励代替简单的二元激励。此时,浊音的激励源是由经过低通滤波的周期脉冲序列与经过高通滤波的白噪声相加而成的,周期脉冲与噪声的混合比例随输入语音的浊化程

度变化。清音的激励源是白噪声加上位置随机的一个正脉冲跟随一个负脉冲的脉冲对形成的爆破脉冲。对于爆破音，脉冲对的幅度增大，与语音的突变成正比。采用混合激励可以使原来二元激励合成引起的金属声、重击声、音调噪声等得到改善。

采用激励脉冲加抖动的方式。将基音相关性不是很强或残差信号中有大的峰值的语音帧判定为抖动的浊音帧。除采用脉冲加噪声的混合激励外，激励信号中的周期脉冲的相位要做随机地抖动，即对每个基音周期的长度乘上一个 0.75～1.25 之间均匀分布的随机数，这样可以改善语音的自然度。

采用单脉冲与码本相结合的激励模式。可取多脉冲激励线性预测编码与码本激励线性预测编码各自的长处，对不同的语音段采用不同的激励模式。对于具有周期性的语音段用以基音周期重复的单脉冲作为激励源，非周期性语音段用从码本中选择的随机序列作为激励源。

(2) 改进基音提取方法

计算线性预测残差信号或者语音信号的自相关函数，并利用动态规划的平滑算法来更准确地提取基音周期。将一帧的线性预测残差信号低通滤波后，求出所有可能的基音时延点上的归一化自相关系数，选出其中 L 个最大值，再用相邻 3 帧的每帧 L 个最大值，用动态规划算法求得最佳基音值。

(3) 选择线谱对参数 LSP 作为声道滤波器的量化参数。

8.6 语音信号混合编码

混合编码是在保留参数编码的技术精华的基础上，引用波形编码准则去优化激励源信号，克服原有波形编码和参数编码的弱点，而吸取它们各自的长处，在 4～16kbit/s 的速率上能够合成高质量语音。其中用到的主要技术就是合成分析技术和感觉加权滤波器，目标是改进激励模型，合成高质语音。

8.6.1 合成分析技术和感觉加权滤波器

近几十年来，人们在 LPC 算法的基础上，对 16kbit/s 以下的高质量语音编码技术进行了广泛深入的研究和实践。在此速率下，能用于残差信号编码的比特数是较少的。若对残差信号进行直接量化并且使残差信号与它的量化值之间的误差达到最小，并不能保证原始语音与重建语音之间的误差最小，而只有采用合成分析法来计算残差信号的编码量化值才能使得重建语音与原始语音的误差最小。换句话说，合成分析法的改进主要就是对激励的改进，它不是寻找与残差信号相匹配的激励，而是寻找给定合成滤波器的最优激励，使其通过合成滤波器时产生的合成语音最接近于原始语音。由于合成滤波器具有递归结构，因此激励信号的每个样点将影响合成语音的许多样点。也就是说，最佳量化模型的选择不是立即决定的，而是要延迟至少几个样点才被决定。因为这种决定依赖于原始语音和合成语音的残差信号，分析过程即包含有合成过程，所以称为“合成分析预测编码”。

感觉加权滤波器的依据是利用人耳听觉的掩蔽效应(Masking Effect)，在语音频谱中能量较高的频段即共振峰处的噪声相对于能量较低频段的噪声而言不易被感知。因此在度量原始语音与合成语音之间的误差时可以计入这一因素，在语音能量较高的频段，允许二者的误差大一些，反之则小一些。为此可以引入一频域感觉加权滤波器 $W(f)$ 来计算二者

的误差，如下所示

$$e=\int_0^{f_s}|S(f)-\hat{S}(f)|^2W(f)\mathrm{d}f \tag{8.56}$$

其中，f_s 是抽样率，$S(f)$，$\hat{S}(f)$分别是原始语音与合成语音的傅里叶变换。不难证明，只要使积分项在整个域内保持常数值，就可以使 e 达到最小值。这样只要在能量最大的的语音频段内使 $W(f)$较小，而能量较小的频段内 $W(f)$较大，这就能抬高前者的误差能量而降低后者的误差能量，为此选取感觉加权滤波器的 Z 域表达式 $W(z)$为

$$W(z)=\frac{A(z)}{A(z/\gamma)}=\frac{1-\sum_{i=1}^{p}a_iz^{-i}}{1-\sum_{i=1}^{p}a_i\gamma^iz^{-i}} \tag{8.57}$$

感觉加权滤波器的特性由预测系数$\{a_i\}$和 γ 来确定。γ 取值在 0～1 之间，由它控制共振峰区域误差的增加。当 $\gamma=1$ 时，$W(z)=1$，此时没有进行感觉加权；当 $\gamma=0$ 时

$$W(z)=1-\sum_{i=1}^{p}a_iz^{-i} \tag{8.58}$$

它等于语音的 p 阶全极点模型谱的倒数，由此得到的噪声频谱能量分布与语音频谱的能量分布是一致的。图 8.11 中示出了一段原始语音的谱，经感觉加权后所得的误差信号的谱以及感觉加权滤波器的频率响应。由图不难看出，感觉加权滤波器的作用就是使实际误差信号的谱不再平坦，而是有着与语音信号谱具有相似的包络形状。这就使得误差度量的优化过程与感觉上的共振峰对误差的掩蔽效应相吻合，产生较好的主观听觉效果。实际听音的结果表明：在 8kHz 采样率下，γ 取值为 0.8 左右较为适宜。注意到加权过程既不会引起位率的增加，也不会增加合成过程的复杂度，它仅使编码器的复杂性有所增加。

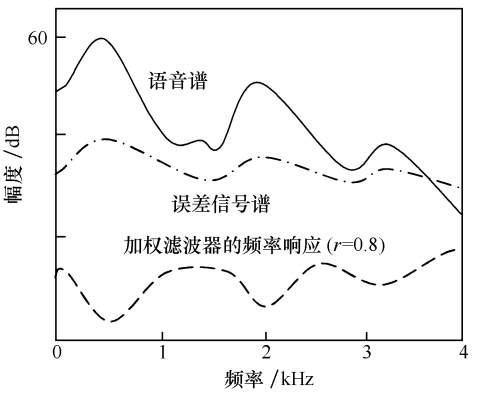


图 8.11　频率响应

8.6.2　激励模型的改进

过于简单的二元激励模型是制约 LPC 编码器声音质量的主要因素。针对此问题，1982 年，Bishnu S. Atal 和 Joel R. Remde 首先提出多脉冲激励线性预测编码(MPE-LPC)算法，在该算法中，每 20ms 语音帧里，传送 16～20 个激励脉冲的位置和幅度信息，能够在 9.6～16kbit/s 速率上，获得相当于 6 位 PCM 编码的质量。1985 年由 Ed. F. Deprettere 和 Perter Kroon 首先提出规则脉冲激励线性预测编码(RPE-LPC)算法，1986 年 K. Hellwig R. Hojmann 和 P. Wary R. J. sluyter 等人在此基础上，改进算法，加入了长时预测 LTP，并使速率降为 13kbit/s，形成长时预测规则脉冲激励(LTP-RPE-LPC)编码方案。它的特点是算法简单，语音质量达到了通信等级。该算法在 1988 年被确定为泛欧标准全速语音编码方案，称为 GSM 标准。1985 年，Manfred R. Schroeder 和 Bishnu S. Atal 首次提出了用矢量量化码本作为激励源的线性预测编码技术 CELP。CELP 以高质量的合成语音及优良的抗噪声和多次转接性能，在 4.8～16kbit/s 速率上得到广泛的应用。1988 年，美国政府采用由美国国防部与 AT&T 贝尔实验室共同研制的 4.8kbit/s CELP 声码器(FED-STD-1016)作为语音编码器标

准;1989 年 8kbit/s 速率的北美数字移动通信全速率编译码器标准采用了修改的 CELP 技术——矢量和激励线性预测编码 VSELP;1991 年 ITU 通过了用短延时码激励线性预测编码 LD-CELP 作为 16kbit/s 语音编码器的 G.728 标准。1996 年 ITU 通过了共轭结构代数码激励线性预测编码器 CS-ACELP 作为 8kbit/s 语音编码器 G.729 标准,这些是码激励的典型算法。

8.6.3 G.728 语音编码标准简介

图 8.12 和图 8.13 分别是 G.728 标准算法中编码器和解码器部分的原理框图。编码部分的工作原理是:首先将速率为 64kbit/s 的 *A*-律或 μ-律 PCM 输入信号转换成均匀量化的 PCM 信号,接着由 5 个连续的语音样点 $s_u(5n)$,$s_u(5n+1)$,…,$s_u(5n+4)$组成一个五维语音矢量 $s(n)=[s_u(5n),s_u(5n+1),\cdots,s_u(5n+4)]$。激励码书中共有 1024 个五维的码矢量。对于每个输入语音矢量,编码器利用合成分析方法从码书中搜索出最佳码矢量,然后将 10bit 的码矢标号通过信道传送给解码器。每 4 个相邻的输入矢量(共 20 个样点)构成一个自适应周期,或者称为帧,每帧更新一次 LPC 系数。因为在 LD-CELP 算法中采用的是后向自适应预测技术,当前的激励增益和综合滤波器的输出是分别对先前量化过的增益和语音信息进行 LPC 分析而得出的,所以向解码器传送的信息只是激励矢量的地址标号,这就使得编码器只有 5 个样点的缓冲延迟,对于 8kHz 的采样率就是 0.625ms 的延迟。把处理延迟和传输延迟包括在内,总的一路编译码延迟不超过 2ms。

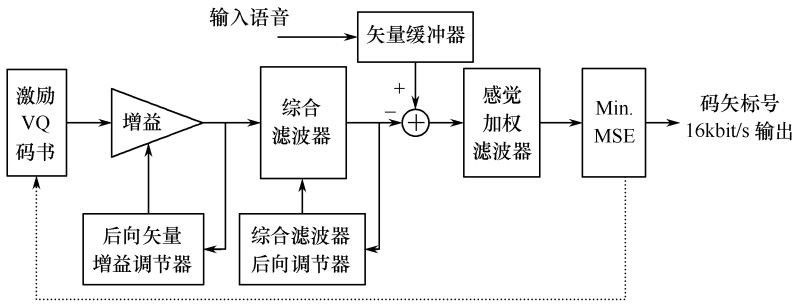


图 8.12　16kbit/s LD-CELP 语音编码器原理框图

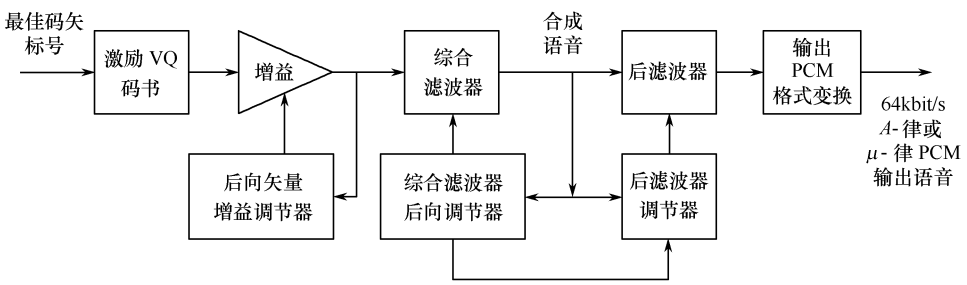


图 8.13　16kbit/s LD-CELP 语音解码器原理框图

解码操作也是逐个矢量地进行的。根据接收到的码矢标号,从激励码书中找到对应的激励矢量,经过增益调整后,得到激励信号,将激励信号输入合成滤波器,就得到合成语音信号。再将合成语音信号进行自适应后滤波处理,以增强语音的主观感觉质量。

8.7 语音信号宽带变速率编码

传统的数字语音通信标准都是基于300～3400Hz的电话带宽，这种窄带语音仅可以保证语音的可理解性，但在语音的自然度及一些特殊音处理方面还不尽人意。近年来一些新兴的应用，如视频会议、第三代移动通信、高保真存储、交互式多媒体服务器等，都要求更大的信号带宽来保持语音的自然度、听觉舒适性以及说话者在特定环境下的现场感。50～7000Hz的语音带宽通常被称为宽带语音频带，包括了人类发声的绝大部分能量范围。同窄带语音相比，宽带语音信号50～300Hz的低频部分增加了语音的自然度、现场感和听觉舒适性，3400～7000Hz的高频部分，可以更好地区分摩擦音，从而增强了语音的可理解性。因此宽带语音不仅提高了语音的可理解性和自然度，而且还增加了透明传输的感觉，使说话方的个人特征体现得更充分。

传统的定速率语音编码从总体上来讲，较高速率的编码算法对语音质量较易保证，但占用网络资源较大；较低速率的编码算法占用网络资源小，但对语音质量较难保证。语音激活检测(VAD)技术的出现和发展，使对有无语音进行判决成为可能，从而可以对背景噪声和激活的语音部分以不同的速率进行编码，降低了平均速率，也就是采用变速率语音编码的方法。人类在进行语音通信时，大约有70%左右的空闲时间没有讲话，始终用一个速率进行语音编码对信道资源是一个浪费。变速率语音编码算法可以根据需要动态调整编码速率，在合成语音质量和系统容量之间取得灵活的折中，最大限度地发挥系统的效能，而且非常适合分组交换网络。

国际标准组织多年来一直在努力定义宽带语音编码标准。早期定义的宽带语音编码标准主要应用于会议电视，近期定义的则主要应用于移动通信和VoIP。宽带语音编码标准G.722、G.722.1及G.722.2(AMR-WB)的详细对比如表8.6所示。

表8.6 宽带语音编码标准对比

标　准	G.722	G.722.1	G.722.2(AMR-WB)
公布时间	1988年	1999年	2002年
编码速率(kbit/s)	64,56,48	32,24	23.85,23.05,19.85,18.25, 15.85,14.25,12.65,8.85,6.60
编码算法	Sub-Band ADPCM	Transform Coder	ACELP
性　能	在64kbit/s接近于透明编码	一些条件下语音质量差，音乐性能较好	12.65kbit/s以上语音质量高 15.85kbit/s与G.722 56kbit/s相当 23.85kbit/s≥G.722 64 kbit/s相当 音乐性能较差
VAD/DTX/CNG	无	无	有
RAM	1KB	2KB	5.3KB
应用	ISDN，视频会议	ISDN，视频会议，VoIP	ISDN，视频会议，VoIP，GSM，WCDMA

G.722是ITU-T 64kbit/s宽带语音编码标准，也是第一个采样率为16kHz的宽带语音编码算法，有三种速率模式，分别为64kbit/s、56kbit/s和48kbit/s，其中64kbit/s速率的语音编码器的MOS值可以达到4.75，它使用了子带－自适应差分脉冲编码SB-ADPCM技术。

G.722 的编码器有两个子带，每个子带的信号用 ADPCM 编码，使用的技术是类似于 G.726 的窄带标准。在编码器端，语音信号以 16kHz 的速率采样，并被分解成相同带宽的两个子带，每个子带的信号在编码前采样速率减半。在解码器端，量化的子带语音信号的采用频率被使用同编码器端分解信号相同的滤波器加倍。重新建立的子带信号被加到一起形成合成信号。

1999 年美国 PictureTel 公司的 Siren 编码算法被 ITU-T 确立为新的宽带语音编码国际标准 G.722.1。G.722.1 主要是为了降低 G.722 的编码速率，可实现比 G.722 编码器更低的比特率以及更大的压缩，它有两种编码速率，分别为 24kbit/s 和 32kbit/s。G.722.1 使用了变换编码技术。

2000 年 12 月，3GPP 选择 AMR-WB 语音编码算法作为第三代移动通信推荐使用的语音编解码算法，于 2001 年 3 月最终确定并正式公布。2002 年 1 月，ITU-T 采纳了 AMR-WB 作为宽带语音编码的新标准，AMR-WB 是通信史上第一种可以同时用于有线与无线业务的语音编码系统。这种算法支持 9 种速率模式(6.6，8.85，12.65，14.25，15.85，18.25，19.85，23.05 和 23.85kbit/s)，相对于 AMR-NB，AMR-WB 语音带宽有所扩展，采样率提升了一倍，音质更加接近面对面交流的效果。

语音编码还有很多方法，这里不一一叙述，有兴趣者可参考相关文献。

第9章　语音合成

9.1　概　述

语音合成是人机语声通信的一个重要组成部分，语音合成技术赋予机器“人工嘴巴”的功能，即解决让机器像人那样说话的问题。

最早的合成器是1835年由W. von Kempelen发明，经Weston改进的机械式会讲话的机器。该机器完全模仿人的发音生理过程，它分别用风箱、特别设计的哨和软管来模拟肺部的空气动力、模拟口腔。而最早的电子式语音合成器也是1939年Homer Dudley发明的声码器，它不是简单地模拟人的生理过程，而是通过电子线路来实现基于语音产生的源－滤波器理论。

但真正具有实用意义的近代语音合成技术是随着计算机技术和数字信号处理技术的发展而发展起来的，主要是让计算机能够产生高清晰度、高自然度的连续语音。在语音合成技术的发展中，早期的研究主要是采用参数合成方法。值得提及的是，1973年Holmes发明的并联共振峰合成器和1980年Klatt发明的串/并联共振峰合成器，只要精心调整参数，这两个合成器都能合成出比较自然的语音。最具代表性的文语转换系统当数美国DEC公司1987年开发的DECtalk。但是，由于准确提取共振峰参数比较困难，虽然利用共振峰合成器可以得到许多逼真的合成语音，但是整体合成语音的音质难以达到文语转换TTS系统的实用要求。

自20世纪80年代末期至今，语音合成技术又有了新的进展，特别是1990年提出的基音同步叠加PSOLA方法，使基于时域波形拼接方法合成的语音的音色和自然度大大提高。20世纪90年代初，基于PSOLA技术的法语、德语、英语、日语等语种的文语转换系统都已经研制成功。这些系统的自然度比以前基于LPC方法或共振峰合成器的文语合成系统的自然度要高，并且基于PSOLA方法的合成器结构简单，易于实时实现，有很大的商用前景。

我国的汉语语音合成研究起步较晚些，但从20世纪80年代初就基本上与国际研究同步发展。大致也经历了共振峰合成、LPC合成到应用PSOLA技术的过程。在国家863计划、国家自然科学基金委、国家攻关计划、中国科学院有关项目等支持下，汉语文语转换系统研究近年来取得了令人瞩目的进展，其中不乏成功的例子：如1993年中国科学院声学所研制的KX-PSOLA，1995年研制的联想佳音；清华大学在1993年研制的TH-SPEECH；1995年中国科技大学研制的KDTALK等系统。这些系统基本上都是采用基于PSOLA方法的时域波形拼接技术，其合成汉语普通话的可懂度、清晰度达到了很高的水平。然而同国外其他语种的文语转换系统一样，这些系统合成的句子及篇章语音机器味较浓，其自然度还不能达到用户可广泛接受的程度，从而制约了这项技术的大规模进入市场。

现阶段语音合成的最大进展是已经能够实时地将任意文本转换成连续可懂的自然语句输出。文语转换使得数据通信和语音通信在终端一级实现交融，人们将有望在获取Internet信息时，使短消息服务、电子邮件等多数以文本方式提供的信息也能用语音的方式输出。语音合

成技术经历了从参数合成到拼接合成，再到两者的逐步结合，其不断发展的动力是人们认知水平和需求的提高。

9.2 语音合成的原理及分类

让机器像人类一样说话，可以仿照人的言语过程模型，设想在机器中首先形成一个要讲的内容，它一般以表示信息的字符代码形式存在；然后按照复杂的语言规则，将信息的字符代码的形式，转换成由基本发音单元组成的序列，同时检查内容的上下文，决定声调、重音、必要的停顿等韵律特性，以及陈述、命令、疑问等语气，并给出相应的符号代码表示。这样组成的代码序列相当于一种“言语码”。从“言语码”出发，按照发音规则生成一组随时间变化的序列，去控制语音合成器发出声音，犹如人脑中形成的神经命令，以脉冲形式向发音器官发出指令，使舌、唇、声带、肺等部分的肌肉协调动作发出声音一样，这样一个完整的过程正是语音合成的全部含义。

实际上，人在发出声音之前要进行一段大脑的高级神经活动，即先有一个说话的意向，然后围绕该意向生成一系列相关的概念，最后将这些概念组织成语句发音输出。

按照人类言语功能的不同层次，语音合成可分成三类层次，如图 9.1 所示。它们是：①按规则从文字到语音的合成（Text-To-Speech）；②按规则从概念到语音的合成（Concept-To-Speech）；③按规则从意向到语音的合成（Intention-To-Speech）。

这三类层次反映了人类大脑中形成说话内容的不同过程，涉及人类大脑的高级神经活动。由于迄今为止我们对人类言语现象的理解仅停留在声道系统的发声过程上，对大脑的高级神经活动还知道得很少，这就使得语音合成的研究，在一段相当长的时期内只能集中于低级阶段，即按规则文－语转换阶段，或者说将书面语言转换成口头语言。这就意味着，目前机器只能达到朗读文章的水平，更高层次的研究还有待于通信、计算机方面的专家和生物学家、语言学家、人工智能专家等的共同努力。

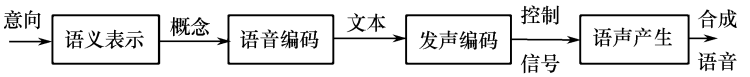


图 9.1 语音合成的各个阶段

语音合成的研究已有多年的历史，现在研究出的语音合成方法的分类，从技术方式讲可分为波形合成法、参数合成法和规则合成方法。

9.2.1 波形合成法

波形合成法一般有两种形式，一种是波形编码合成，它类似于语音编码中的波形编解码方法，该方法直接把要合成的语音的发声波形进行存储或者进行波形编码压缩后存储，合成重放时再解码组合输出。这种语音合成器只是语音存储和重放的器件。其中最简单的就是直接进行 A/D 变换和 D/A 变换，或称为 PCM 波形合成法。显然，用这种方法合成语音，词汇量不可能很大，因为所需的存储容量太大。波形合成法是一种相对简单的语音合成技术，通常只能合成有限词汇的语音段，目前许多专门用途的语音合成器都采用这种方式，如自动报时、报站和报警等。另一种是波形编辑合成，它把波形编辑技术用于语音合成，通过选取音库中采取自然语言的合成单元的波形，对这些波形进行编辑拼接后输出。它采用语音编码技术，存储适当的

语音基元，合成时，经解码、波形编辑拼接、平滑处理等输出所需的短语、语句或段落。与规则合成方法不同，这类方法在合成语音段时所用的基元并不做大的修改，最多只是对相对强度和时长做一点简单的调整。因此这类方法必须选择比较大的语音单位作为合成基元，如选择词、词组、短语、甚至语句作为合成基元，这样在合成语音段时基元之间的相互影响很小，容易达到很高的合成语音质量。

9.2.2 参数合成法

参数合成法也称为分析合成法，是一种比较复杂的方法。为了节约存储容量，必须先对语音信号进行分析，提取出语音的参数，以压缩存储量，然后由人工控制这些参数的合成。参数合成法一般有发声器官参数合成和声道模型参数合成。发声器官参数合成法是对人的发声过程直接进行模拟。它定义了唇、舌、声带的相关参数，如唇开口度、舌高度、舌位置、声带张力等，由发声参数估计声道截面积函数，进而计算声波。由于人的发音生理过程的复杂性和理论计算与物理模拟的差别，合成语音的质量不理想。声道模型参数语音合成是基于声道截面积函数或声道谐振特性合成语音的。早期语音合成系统的声学模型，大多通过模拟人的口腔的声道特性来产生。后来又产生了基于 LPC、LSP 等声学参数的合成系统。这些方法用来建立声学模型的过程为：首先录制声音，这些声音涵盖了人发音过程中所有可能出现的读音；提取出这些声音的声学参数，并整合成一个完整的音库。在发音过程中，首先根据需要发的音，从音库中选择合适的声学参数，然后根据韵律模型中得到的韵律参数，通过合成算法产生语音。参数合成方法的优点是其音库一般较小，并且整个系统能适应的韵律特征的范围较宽，这类合成器比特率低，音质适中；缺点是参数合成技术的算法复杂，参数多，并且在压缩比较大时，信息丢失也大，合成出的语音总是不够自然、清晰。为了改善音质，近几年发展了混合编码技术，主要是为了改善激励信号的质量，虽然比特率有所增大，但音质得到了提高。

9.2.3 规则合成法

这是一种高级的合成方法。规则合成方法通过语音学规则产生语音。合成的词汇表不是事先确定的，系统中存储的是最小的语音单位的声学参数，以及由音素组成音节、由音节组成词、由词组成句子和控制音调、轻重音等韵律的各种规则。给出待合成的字母或文字后，合成系统利用规则自动地将它们转换成连续的语音声波。这种方法可以合成无限词汇的语句。这种算法中，用于波形拼接和韵律控制的较有代表性的算法是 20 世纪 80 年代末，由 F. CharPentier 等人提出的基音同步叠加 PSOLA 技术，该方法既能保持所发音的主要音段特征，又能在拼接时灵活调整其基频、时长和强度等韵律特征。其主要特点是：在语音波形片断拼接之前，首先根据语义，用 PSOLA 算法对拼接单元的韵律特征进行调整，使合成波形既保持了原始语音基元的主要音段特征，又使拼接单元的韵律特征符合语义，从而获得很高的可懂度和自然度。在对拼接单元的韵律特征进行调整时，它以基音周期（而不是传统的定长的帧）为单位进行波形的修改，把基音周期的完整性作为保证波形及频谱的平滑连续的基本前提。PSOLA 算法使语音合成技术向实用化迈进一大步。当前，越来越多的人研究波形拼接语音合成技术，并设计了相应的算法和系统。国内的文语转换系统，也主要采用基于 PSOLA 方法的语音合成技术。汉语音节的独立性较强，音节的音段特征比较稳定，但汉语音节的音高、音长和音强等韵律特征在连续语流中变化复杂，而这些韵律特征又是影响汉语合成语音自然度的主要因素。因此，汉语很适合采用基于 PSOLA 技术的波形拼接法来合成。表 9.1 列出了

三种语音合成方式的特征比较。

表 9.1　三种语音合成方式的比较

项目		波形合成方式	参数合成方式	按规则合成方式
语音质量	可懂度	高	高	中
	自然度	高	中	低
词汇量		小(500 字以下)	大(数千字)	无限
合成方法		PCM,ADPCM	LPC,LSP,共振峰	LPC,LSP 共振峰
数码率		9.6～64kbit/s	2.4～9.6kbit/s	50～75kbit/s
1Mb 可合成的语音长度		15～100 秒	100 秒～7 分	无限
合成基元		音节、词组、句子	音节、词组、句子	音素、双音素、音节
装置		简单	比较复杂	复杂
硬件主体		存储器	存储器和处理器	处理器

9.3　共振峰合成法

共振峰合成是目前一种比较成熟的参数合成方法。其理论基础是语音生成的数学模型。在该模型中,语音生成过程是在激励信号的激励下,经过谐振腔(声道),由口或鼻腔辐射声波。因此,声道参数、声道谐振特性一直是研究的重点。共振峰合成模型是把声道视为一个谐振腔,利用腔体的谐振特性,如共振峰频率及带宽,以此为参数构成一个共振峰滤波器。因为音色各异的语音有不同的共振峰模式,以每个共振峰频率及其宽带为参数,可以构成一个共振峰滤波器。将多个这种滤波器组合起来模拟声道的传输特性,对激励声源发生的信号进行调制,经过辐射即可得到合成语音。这便是共振峰语音合成器的构成原理。实际上,共振峰滤波器的个数和组合形式是固定的,只是共振峰滤波器的参数,随着每一帧输入的语音参数而改变,以此表征音色各异的语音的不同的共振峰模式。基于共振峰的理论有以下三种实用模型。

9.3.1　级联型共振峰模型

对于一般元音,其共振峰特性可以用一个全极点模型来描述,每对极点表示一个共振峰,而每对共轭极点可以用一个二阶滤波器实现,因此用多个二阶滤波器级联就可以实现整个模型。在该模型中,声道被认为是一组串联的二阶谐振器,共振峰滤波器首尾相接,其传递函数为各个共振峰的传递函数相乘的结果。如图 9.2 所示就是有 5 个极点的共振峰级联模型,其传递函数为

$$v(z)=\frac{G}{1-\sum_{k=1}^{10}a_k z^{-k}} \tag{9.1}$$

即

$$v(z)=G\cdot\prod_{i=1}^{5}v_i(z)=G\cdot\prod_{i=1}^{5}\frac{1}{1-b_i z^{-1}-c_i z^{-2}} \tag{9.2}$$

式中,G 为增益因子。

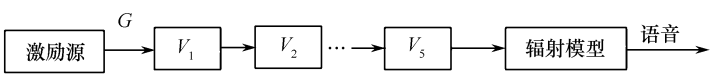


图 9.2　共振峰级联模型

9.3.2 并联型共振峰模型

对于鼻化元音等非一般元音以及大部分辅音，上述级联型模型不能很好地加以描述和模拟，因此，产生了并联型共振峰模型。在并联型模型中，输入信号先分别进行幅度调节，再加到每一个共振峰滤波器上，然后将各路的输出叠加起来。其传递函数为

$$v(z)=\frac{\sum_{r=0}^{R}b_r z^{-r}}{1-\sum_{k=1}^{p}a_l z^{-k}} \tag{9.3}$$

上式可分解成部分分式之和

$$v(z)=\sum_{l=1}^{M}\frac{A_l}{1-B_l z^{-1}-C_l z^{-2}} \tag{9.4}$$

其中，A_l 为各路的增益因子。图 9.3 就是一个 $M=5$ 的并联型共振峰模型。

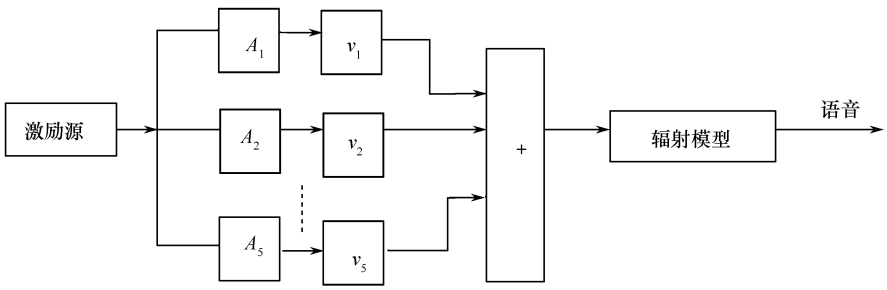


图 9.3　并联型共振峰模型

9.3.3 混合型共振峰模型

比较以上两种模型，对于大多数的元音，级联型合乎语音产生的声学理论，并且无须为每一个滤波器分设幅度调节；而对于大多数清擦音和塞音，并联型则比较合适，但是其幅度调节很复杂。于是考虑将两者结合在一起，提出了混和型共振峰模型，如图 9.4 所示。

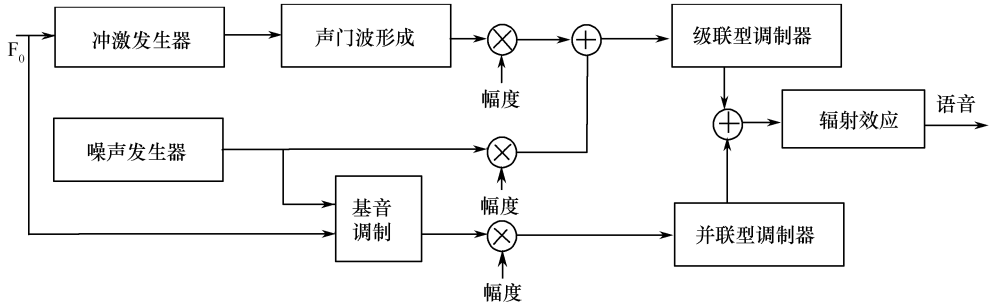


图 9.4　混和型共振峰模型

对于共振峰合成器的激励，简单地将其分为浊音和清音两种类型是有缺陷的，因为对浊辅音，尤其是其中的浊擦音，声带振动产生的脉冲波和湍流同时存在，这时噪声的幅度要被声带

振动周期性的调制。因此，为了得到高质量的合成语音，激励源应具备多种选择，以适应不同的发音情况。图 9.4 中激励源有三种类型：合成浊音语音时用周期冲激序列；合成清音语音时用伪随机噪声；合成浊擦音语音时用周期冲激调制的噪声。激励源对合成语音的自然度有明显的影响。发浊音时，最简单的是三角波脉冲，但这种模型不够精确，对于高质量的语音合成，激励源的脉冲形状是十分重要的，可以采用其他更为精确的形式。合成清音时的激励源一般使用白噪声，实际实现时用伪随机数发生器来产生。

共振峰模型是基于对声道的一种比较准确的模拟，因而可以合成出自然度相对较高的语音，另外，由于共振峰参数有着明确的物理意义，直接对应于声道参数，因此，可以比较容易地利用共振峰描述自然语流中的各种现象，总结出声学规则，最终用于共振峰合成系统。但是，共振峰合成技术也有明显的弱点。首先由于它是建立在对声道的模拟上，因此，声道模型的不精确势必会影响其合成质量。另外，实际工作中共振峰模型虽然描述了语音中最基本最主要的部分，但并不能表征影响语音自然度的其他许多细微的语音成分，从而影响了合成语音的自然度。其次，共振峰合成器控制十分复杂，对于一个好的合成器来说，其控制参数往往达到几十个，实现起来十分困难。

一般的共振峰合成器模型中，声源和声道间是互相独立的，没有考虑它们之间的相互作用。然而，研究表明，在实际语言产生的过程中，声源的振动对声道里传播的声波有不可忽略的作用。因此提高合成音质的一个重要途径，还必须采用更符合语音产生机理的语音生成模型。高级共振峰合成器可合成出高质量的语音，几乎和自然语音没有差别，因此，长期以来，共振峰合成器也一直处于主流地位。但关键是如何得到合成所需的控制参数，如共振峰频率、带宽、幅度等。而且，求取的参数还必须逐帧修正，才能使合成语音与自然语音达到最佳匹配。在以音素为基元的共振峰合成中，可以存储每个音素的参数，然后根据连续发音时音素之间的影响，从这些参数内插得到控制参数轨迹。尽管共振峰参数理论上可以计算，但实验表明，这样产生的合成语音在自然度和可懂度方面均不令人满意。理想的方法是从自然语音样本出发，通过调整共振峰合成参数，使合成出的语音和自然语音样本在频谱的共振峰特性上最佳匹配，即误差最小，此时的参数作为控制参数，这就是合成分析法。

9.4 线性预测参数合成法

线性预测参数合成法是目前比较简单和实用的一种语音合成方法，以其低数据率、低复杂度、低成本，受到特别的重视。20 世纪 60 年代后期发展起来的线性预测编码(LPC)语音分析方法可以有效地估计基本语音参数，如基音、共振峰、谱、声道面积函数等，可以对语音的基本模型给出精确的估计，而且计算速度较快。

线性预测合成方法是目前比较简单和实用的一种语音合成方法，它以其低数据率、低复杂度、低成本，受到特别的重视。20 世纪 60 年代后期发展起来的 LPC 语音分析方法可以有效地估计基本语音参数，如基音、共振峰，谱、声道面积函数等，可以对语音的基本模型给出精确的估计，而且计算速度较快。因此，LPC 语音合成器利用 LPC 语音分析方法，通过分析自然语音样本，计算出 LPC 系数，就可以建立信号产生模型，从而合成出语音。线性预测合成模型是一种“源滤波器”模型，由白噪声序列和周期脉冲序列构成的激励信号，经过选通、放大并通过时变数字滤波器(由语音参数控制的声道模型)，就可以再获得原语音信号。这种参数编码的语音合成器的框图如图 9.5 所示。

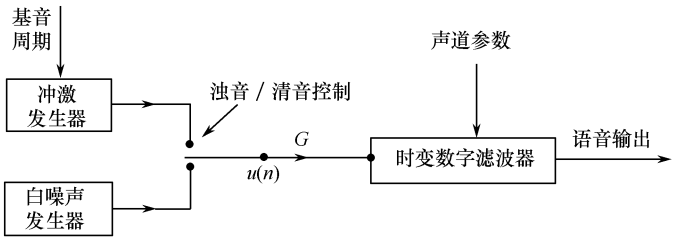


图 9.5　LPC 语音合成器的框图

图 9.5 所示的线性预测合成的形式有两种：一种是直接用预测器系数 a_i 构成的递归型合成滤波器，其结构如图 9.6 所示，用这种方法定期地改变激励参数 $u(n)$和预测系数 a_i，就能合成出语音。这种结构简单而直观，为了合成一个语音样本，需要进行 p 次乘法和 p 次加法。它合成的语音样本由下式决定

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n) \tag{9.5}$$

其中，a_i 为预测系数；G 为模型增益；$u(n)$为激励；合成样本为 $s(n)$；p 为预测器阶数。

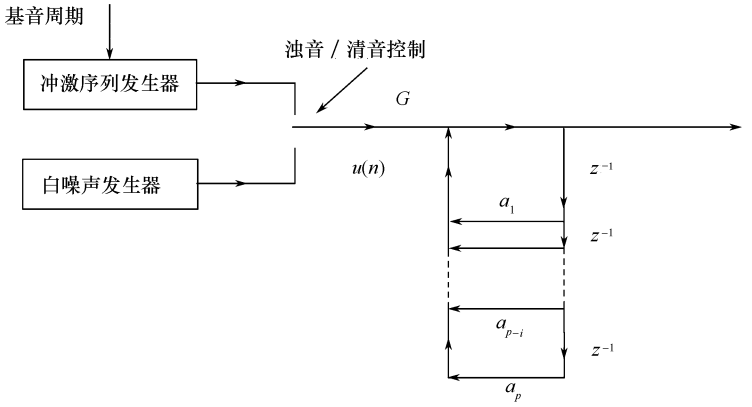


图 9.6　直接用预测器系数 a_i 构成的合成滤波器

直接形式的预测系数滤波器结构的优点是简单、易于实现，所以曾经被广泛采用。其缺点是合成语音样本需要很高的计算精度。这是因为这种递归结构对系数的变化非常敏感，其系数的微小变化就可以导致滤波器极点位置的很大变化，甚至出现不稳定现象。所以，由于预测系数 a_i 的量化所造成的精度下降，使得合成的信号不稳定，容易产生振荡的情况。而且预测系数的个数 p 变化时，系数 a_i 的值的变化也很大，很难处理，这是直接形式的线性预测法的缺点。

另一种合成的形式是采用反射系数 k_i 构成的格型合成滤波器。它的合成语音样本由下式决定

$$s(n) = Gu(n) + \sum_{i=1}^{p} k_i b_{i-1}(n-1) \tag{9.6}$$

其中，G 为模型增益；$u(n)$为激励；k_i 为反射系数；$b_i(n)$为后向预测误差；p 为预测器阶数。

由式(9.6)可以看出，只要知道反射系数，激励位置(即基音周期)和模型增益就可以由后向误差序列迭代计算出合成语音。合成一个语音样本需要$(2p-1)$次乘法和$(2p-1)$次加法。采用反射系数 k_i 的格型合成滤波器结构，虽然运算量大于直接型结构，却具有一系列优点：其

参数 k_i 具有 $|k_i|<1$ 的性质，因而滤波器是稳定的；同时与直接结构形式相比，它对有限字长引起的量化效应灵敏度较低。此外，基音同步合成需要对控制参数进行线性内插，以得到每个基音周期的起始处的值。然而预测器系数本身却不能直接内插，但可以证明，对于部分相关系数进行内插，如果原来的参数是稳定的，则结果必定稳定。无论选用哪一种滤波器结构形式，LPC 合成模型中所有的控制参数都必须随时间不断修正。

在实际进行语音合成时，除了构成合成滤波器之外，还必须有激励信号作为音源。在合成浊音的情况下，要有一定基音周期的脉冲序列作为音源；在合成清音的情况下，将白噪声作为音源。同时，还需要进行清/浊音判别并确定音源强度。

普通的线性预测编码方式存在一些不足，对于共振峰的音节，如鼻音和鼻化元音，很难被模拟，对于短的爆破音，由于时域长度可能比用于分析的帧长更短，模拟质量不好。因此，用标准的 LPC 方法合成语音的质量通常很差。但是，在对基本模型做出一些改进后，合成质量可以被提高。为此，在基本的 LPC 模型基础上发展了其他的线性预测方法，如由误差信号作为激励信号的残差激励线性预测(RELP)等，这些方法中的激励信号与普通的 LPC 语音合成方法中的激励信号相比有所改进，可以更准确地合成出语音信号。近些年，在利用 LPC 合成技术进行语音合成的基础上，又引进了多脉冲激励 LPC(MPE-LPC)技术，矢量量化技术(VQ)，码激励(CELP)技术，这些技术对于 LPC 合成技术的应用起了很大的作用，进一步提高了 LPC 语音合成法的应用效果和领域。

LPC 语音合成和共振峰语音合成是两种经典的语音合成技术，因此有必要对这两种技术做一个归纳性的比较：

① LPC 语音合成有比较简单和完全自动的分析步骤，合成器结构也比较简单，采用格形滤波器时，量化特性和稳定性都比较好，硬件实现容易；而共振峰合成需要较多的参数调整，合成器结构相对讲要复杂些。

② 共振峰合成原理和实际发声原理联系紧密，它的模型控制参数对合成语音谱特性的影响比较直观。基于我们对人类发声的了解，容易确定语音合成所需要的参数变化轨迹以及在语音段边界处的参数内插。在 LPC 合成中，控制 LPC 系数的变化轨迹是十分有限的，因为合成语音频谱特性由系数多项式决定，每一个系数都在一个宽的范围内，以相当复杂的方式影响着合成语音的频谱特性，很难找出简便的调整方法。

③ 共振峰语音合成比较灵活，允许简单地变换以模仿不同人的发音，通过共振峰频率的移动，容易改变语声中和讲话人特征有关的部分；而 LPC 合成则比较困难，只有将 LPC 的反射系数转变成极点的位置，才有可能做类似的修正。

④ 由于线性预测方法对谱包络的谷点的模型要比峰点差得多，因此共振峰带宽的估计一般是不合适的；而共振峰合成方法中，共振峰的带宽还可以从离散傅里叶变换谱来估计，尽管也有一定的困难，但相对来说，带宽的估计要正确些。

⑤ 标准 LPC 的全极点模型，对具有零点谱特性的那些音，特别是鼻音，效果比较差；共振峰合成方法则可以采用反谐振器来直接模拟鼻音中最重要的频谱零点，使得合成语音音质得以提高。

⑥ 从总体上说，选择 LPC 语音合成还是共振峰合成，基于两个因素的折中；LPC 合成具有简单，可自动进行系数分析的优点；而比较复杂的共振峰合成可望产生较高质量的合成语音。

9.5 基音同步叠加法

基音同步叠加 PSOLA 算法是一种波形编辑技术，其核心思想是直接对存储于音库中的语音运用 PSOLA 算法进行拼接，从而整合成完整的语音。有别于传统概念中只是将不同的语音单元进行简单拼接，该系统首先要在大量语音库中，选择最合适的语音单元用于拼接，并且在选择语音单元的过程中往往采用多种复杂的技术，最后在拼接时，使用 PSOLA 算法，根据上下文的要求，对其合成语音的韵律特征进行修改。而且，音库中的采样波形保留了一部分原发音人的语音特征，这样使合成语音的自然度和清晰度都得到了显著提高。

决定语音波形韵律的主要时域参数包括音长、音强、音高等。音长的调节对于稳定的波形段是比较简单的，只需以基音周期为单位加/减波形即可。但由于语音单元本身的复杂性，实际处理时采用特定的时长缩放法；音强对应于语音波形的幅度，音强改变只要加权波形数据即可，但对一些重音有变化的音节，其幅度包络也需要改变；音高的大小对应于波形的基音周期。对大多数通用语言，音高仅代表语气的不同和说话人的更替，但汉语的音高曲线构成声调，声调具有区分语义作用，因此汉语的音高修改比较复杂。

由于韵律修改所针对的侧面不同，PSOLA 算法的实现目前有三种方式。分别为时域基音同步叠加 TD-PSOLA、线性预测基音同步叠加 LPC-PSOLA 和频域基音同步叠加 FD-PSOLA。其中 TD-PSOLA 算法计算效率较高，已被广泛应用，是一种经典算法，这里只介绍 TD-PSOLA 算法原理。

9.5.1 基音同步叠加 PSOLA 算法原理

PSOLA 法来源于利用短时傅里叶变换重构信号的叠接相加法。信号 $x(n)$的短时傅里叶变换为

$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{+\infty} x(m)w(n-m)e^{-j\omega m}, \quad n \in Z \tag{9.7}$$

其中，$w(n)$是长度为 N 的窗序列，Z 表示全体整数集合。注意到 $X_n(e^{j\omega})$是变量 n 和 ω 的二维时频函数，对于 n 的每个取值都对应有一个连续的频谱函数，显然存在较大的信息冗余，所以可以在时域每隔若干个（例如 R 个）样本取一个频谱函数就可以重构原信号 $x(n)$。令：

$$Y_r(e^{j\omega}) = X_n(e^{j\omega})|_{n=rR}, \quad r,n \in Z \tag{9.8}$$

其傅里叶逆变换为

$$y_r(m) = \frac{1}{2\pi}\int_{-\infty}^{\infty} Y_r(e^{j\omega})e^{j\omega m}d\omega, \quad m \in Z \tag{9.9}$$

然后将 $y_r(e^{j\omega})$叠接相加便可得

$$y(m) = \sum_{r=-\infty}^{\infty} y_r(m) = \sum_{r=-\infty}^{\infty} x(m)w(rR-m) = x(m)\sum_{r=-\infty}^{\infty} w(rR-m) \quad m \in Z \tag{9.10}$$

通常选 $w(n)$是对称的窗函数，所以有 $w(rR-n)=w(n-rR)$，可以证明，对于汉明窗来说，当 $R \leqslant N/4$ 时，无论 m 为何值都有

$$\sum_{r=-\infty}^{\infty} w(rR-m)=\frac{W(\mathrm{e}^{\mathrm{j}0})}{R} \tag{9.11}$$

所以

$$y(n)=x(n)\cdot\frac{W(\mathrm{e}^{\mathrm{j}0})}{R} \tag{9.12}$$

其中，$W(\mathrm{e}^{\mathrm{j}\omega})$为$w(n)$的傅里叶变换。式(9.12)说明，用叠接相加法重构的信号$y(n)$与原信号$x(n)$只相差一个常数因子。

在这里讨论叠接相加法的目的不是为了完全重构原信号，而是要对原信号进行基频、时长、短时能量等韵律特征的修改，使信号的动态谱包络不发生大的改变。这涉及到在合成信号时，是采取波形逼近还是谱包络逼近的原则问题。波形逼近，实际上就是对信号进行重构，它所能提供的韵律调整余地较小；谱包络逼近，虽然失掉了相位信息，但获得了较大的调整空间，且人耳对于声波的相位感知并不灵敏。这里采用原始信号谱与合成信号谱均方误差最小的叠接相加合成公式。定义两信号$x(n)$和$y(n)$之间谱距离测度

$$D[x(n),y(n)]=\sum_{t_g}\frac{1}{2\pi}\int_{-\pi}^{\pi}|X_{t_m}(\mathrm{e}^{\mathrm{j}\omega})-Y_{t_g}(\mathrm{e}^{\mathrm{j}\omega})|^2\mathrm{d}\omega \tag{9.13}$$

其中，$X_{t_m}(\mathrm{e}^{\mathrm{j}\omega})$为$n=t_m$处的加窗短时信号$w_1(n-t_m)x(n)$的短时傅里叶变换；$Y_{t_g}(\mathrm{e}^{\mathrm{j}\omega})$为$n=t_g$处的加窗短时信号$w_2(n-t_g)y(n)$的短时傅里叶变换；$\{t_m\}$和$\{t_g\}$分别为$x(n)$和$y(n)$的基音标注点，是一系列与基音同步的在信号时间轴上的标注点，可以取每个基音周期中信号绝对值为最大值的位置。$w_1(n-t_m)x(n)$即是与t_m同步的短时信号。为了得到合成信号，将$w_1(n-t_m)x(n)$调整成为与t_g同步的短时信号$w_2(n-t_g)y(n)$时，是按韵律规则进行的。根据移位定理和Parseval定理，上式可改写为

$$\begin{aligned}D[x(n),y(n)]&=\sum_{t_g}\sum_{n=-\infty}^{\infty}\{w_1[t_m-(n+t_m)]x(n+t_m)-w_2[t_g-(n+t_g)]y(n+t_g)\}^2\\&=\sum_{t_g}\sum_{n=-\infty}^{\infty}[w_1(n+t_g)x(n+t_g+t_m)-w_2(n+t_g)y(n)]^2\end{aligned} \tag{9.14}$$

要求合成信号$y(n)$满足谱距离$D[x(n),y(n)]$最小，可以令

$$\frac{\partial D[x(n),y(n)]}{\partial y(n)}=0 \tag{9.15}$$

解得

$$y(n)=\frac{\sum_{t_g}w_1(n+t_g)w_2(n+t_g)x(n+t_g+t_m)}{\sum_{t_g}w_2^2(n+t_g)} \tag{9.16}$$

窗函数$w_1(n)$和$w_2(n)$可以是两种不同的窗函数，其长度也可以不相等。式(9.16)就是在谱均方误差最小意义下的时域基音同步叠接相加合成公式。从此式可以看出，如果原信号是与$\{t_m\}$为基音同步的短时信号的叠加，合成后的信号就变成了式(9.16)所表示的与$\{t_g\}$为基音同步的短时信号的叠加，而这时引入的谱失真量是最小的。

实际合成时$w_1(n)$和$w_2(n)$可以用完全相同的窗，分母可视为常数，而且可以加一个短时幅度因子α_{t_g}来调整短时能量，即

$$y(n)=\frac{\sum_{t_g}\alpha_{t_g}w_1(t_g-n)w_2(t_g-n)x(n-t_g+t_m)}{\sum_{t_g}w_2^2(t_g-n)} \tag{9.17}$$

当窗长取为对应目标基音周期的 2 倍时，可取 $\alpha_{t_g}=1$。

基音同步叠接相加法是具有良好的韵律调整能力的，但也有不足之处，当基音频率修改过大时有可能出现严重的谱包络失真，即共振峰特性产生不可接受的变异。

9.5.2 基音同步叠加 PSOLA 算法实现步骤

概括起来说，用 PSOLA 算法实现语音合成时主要有三个步骤。分别为基音同步分析、基音同步修改和基音同步合成。下面介绍这三个步骤。

1. 基音同步分析

同步标记是与合成单元浊音段的基音保持同步的一系列位置点，用它们来准确反映各基音周期的起始位置。同步分析的功能主要是对语音合成单元进行同步标记设置。PSOLA 技术中，短时信号的截取和叠加，时间长度的选择，均是依据同步标记进行的。对于浊音段有基音周期，而清音段信号则属于白噪声，所以这两种类型需要区别对待。在对浊音信号进行基音标注的同时，为保证算法的一致性，一般令清音的基音周期为一常数。以语音合成单元的同步标记为中心，选择适当长度（一般取两倍的基音周期）的时窗对合成单元做加窗处理，获得一组短时信号 $x_m(n)$

$$x_m(n)=w_m(t_m-n)x(n) \tag{9.18}$$

其中，t_m 为基音标注点，$w_m(n)$ 一般取汉明窗，窗长大于原始信号的一个基音周期，因此窗间有重叠。窗长一般取为原始信号的基音周期的 2～4 倍。

2. 基音同步修改

同步修改在合成规则的指导下，调整同步标记，产生新的基音同步标记。具体地说，就是通过对合成单元同步标记的插入、删除来改变合成语音的时长；通过对合成单元标记间隔的增加、减小来改变合成语音的基频等。这些短时合成信号序列在修改时与一套新的合成信号基音标记同步。在 TD-PSOLA 方法中，短时合成信号是由相应的短时分析信号直接复制而来。若短时分析信号为 $x(t_a(s),n)$，短时合成信号为 $x(t_s(s),n)$，则有

$$x(t_a(s),n)=x(t_s(s),n) \tag{9.19}$$

式中，$t_a(s)$ 为分析基音标记，$t_s(s)$ 为合成基音标记。

3. 基音同步合成

基音同步合成是利用短时合成信号进行叠加合成。如果合成信号仅仅在时长上有变化，则增加或减少相应的短时合成信号；如果是基频上有变化，则首先将短时合成信号变换成符合要求的短时合成信号再进行合成。

基音同步叠加合成的方法有很多，这里使用前面给出的式(9.17)。利用式(9.17)，可以通过对原始语音的基音同步标志 t_m 间的相对距离的伸长和压缩，对合成语音的基音进行灵活的提升和降低，同样还通过对音节中的基音同步标志的插入和删除来实现对合成语音音长的改变，最终得到一个新的合成语音的基音同步标志 t_g，并且可以通过对式(9.17)中能量因子 a_{tg} 的变化来调整语流中不同部位的合成语音的输出能量。图 9.7 所示即为同步叠加算法改变语音基音和时长的示意图。

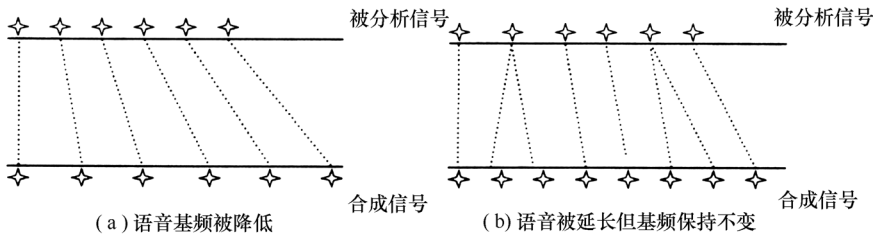


图 9.7　时域基频同步合成语音

9.6　文语转换系统

9.6.1　文语转换系统的组成

为了使文语转换 TTS 系统输出的语音清晰、自然、流畅，系统中应当具有一个性能优良的语音合成模块。但是仅仅将一个个单字的发音机械地连接起来，这样合成的语音缺乏自然度。语音的自然度取决于其发音声调的变化，而在连续语流中一个字的发音不仅与这个字本身的发音有关，而且还要受到它前后与其相邻字的发音的影响。所以在文语转换系统中，必须事先对文本进行分析，根据上下文的关系来确定每个字发音的声调应如何变化，然后用这些声调变化参数去控制语音的合成。因此，文语转换系统还应当具有文本分析和韵律控制功能的模块。文本分析、韵律控制和语音合成这三个模块是文语转换系统的三个核心部分，其结构如图 9.8 所示。

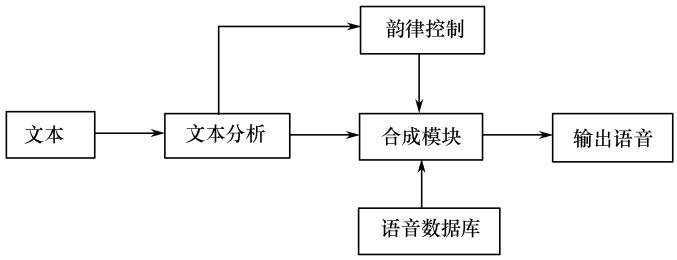


图 9.8　TTS 系统基本框图

1. 文本分析

文本分析的主要功能是使计算机能够识别文字，并根据上下文关系在一定程度上对文本进行理解，从而知道要发什么音、怎么发音，并将发音的方式告诉计算机，另外还要让计算机知道文本中哪些是词，哪些是短语、句子，发音时应该停顿的位置和时长等。文本分析的工作过程包括：① 将输入的文本规范化，在这个过程中处理用户可能的拼写错误，并将文本中出现的一些不规范或无法发音的字符过滤掉；② 分析文本中的词或短语的边界，确定文字的读音，同时在这个过程中分析文本中出现的数字、姓氏、特殊字符以及各种多音字的读音方式；③ 根据文本的结构、组成和不同位置出现的标点符号，来确定发音时语气的变换以及不同音的轻重方式。最终，文本分析模块将输入的文字转换成计算机能够处理的内部参数，便于后续模块进一步处理并生成相应的信息。

2. 韵律控制

任何人说话都有韵律特征，有不同的声调、语气、停顿方式，发音长短也各不相同，这些都

属于韵律特征。而韵律参数则包括了能影响这些特征的声学参数，如基频、音长、音强等。最终系统能够用来进行语音信号合成的具体韵律参数，还要靠韵律控制模块。

3. 语音合成

文语转换系统的合成语音模块一般采用波形拼接来合成语音的方法，其中最具代表性的是前面介绍过的基音同步叠加法 PSOLA。

9.6.2 汉语按规则合成

通过语音学规则产生语音，对于不同的语种，其规则是完全不同的，这里仅讨论文语转换层次上的汉语按规则合成中有关韵律规则的几个基本问题。

文语转换系统首先接收键盘或文件按一定格式输入的文本信息；然后按照给定的语言学规则决定各字的发音（合成）基元序列，以及基元组合时的韵律特性，如音长、重音、声调、语调等；从而决定为合成整个文本所需的“言语码”；最后再用这些代码控制机器去语音库中取出相应的语音参数，进行合成运算，得到语音输出。汉语语音属于声调语言，有复杂的韵律结构。汉语语句结构中的语音层次为：音素→音节→词语→句子。声学基元是指拼接的基本单位，它可能是音素、双音素、三音素、半音节（首音、尾音）、音节、词语和语句等。基元越小，语音数据库越小，拼接越灵活，韵律特征的变化就越复杂。按规则合成无限词汇的汉语语音时，基元的选择一般应选声母和韵母。如果选择音素为基元，虽然其存储量可以做到很小，但是汉语中音素的音位变体规律非常复杂。因此，在汉语语音合成中，采用音素或双音素作为基元是不合适的。另一方面，如果采用音节甚至采用单词为合成基元，虽然这时所需的规则要简单些，但是语音库的存储容量要大大增加。折中考虑，一般采用声母与韵母作为合成基元，存储容量不大，而所需的规则大体上只是：“辅音→元音、元音→元音转接规则”和“多字词中各字的声调变调规则”等。与其他合成技术相比较，规则合成有两个明显的优点：语音库占用的内存很小；可以灵活控制合成语音的声学特征和韵律特征。

韵律规则是合成规则中的一个重要组成部分。语流中的抑扬顿挫、轻重相随、节奏分明，就是由音高、音长和强度等方面的变化所表现出来的特征，称为“韵律特征（prosodic feature）”，也叫“超音段特征”。它们反映了语音在基频、共振峰、能量及谱分布特性上的差异。对于同一个基元，由于语境不同和重音的表现不同，其声学特征有很大的差别。通过对语音数据的声学参数，如基频、音长、音强等修改，可以进行重音、语调的模拟，实现语速、调高的变化。韵律特征主要包括声调、语调、重音等。声调属于音节层的韵律；语调属于句子层的韵律。韵律对合成语音的自然度、可懂度及流畅性影响极大。

1. 重音规则

重音在语言交流中起到重要作用。一般说汉语的重音，是指说话或朗读时读的比较重的音节或词语。然而，汉语的重音并不像非声调的重音那样说的声大一点，用劲一点，而是要时间长一点，音程大一点，也就是使低的更低，高的更高。一般可以将汉语重音分为词重音和句重音两大类。所谓词重音，指词的某个音节可分为重轻等级。音长特征是区分这个等级的主要标志，轻声的音长较短。另外一个重要的区分特征是声调域，轻声的声调域缩小，这就使轻声字所需的能量减少，但强度并不一定减弱。汉语重音的声学特征表现在音域加宽、音程加大，气流加强。

2. 转接与音渡规则

转接与音渡是音素序列转变成语音流时的动态变化规律。人在说话时，发声器官的运动

是连续的，而声道的形状不可能突变。因此连续语音流决不是相邻的各音素简单的组合和拼接，它们之间有着不同程度的相互影响。特别当发音速度较快时，前一个音素还没有发完，舌、口、唇等已经向下一个位置移动，准备或开始发下一个音了。由于实际发音时牵涉到各个发声器官，所以音素之间的过渡现象十分复杂。在汉语发音中，存在两种基本的过渡，即辅音与元音组合和元音与元音组合。前者出现在声母和韵母的拼接过程中，称为“转接”；后者出现在复合韵母内部，称为“音渡”。

所谓转接是指前一个辅音对其后元音共振峰的影响。同一元音的共振峰特性受不同辅音的影响会有很大的变化，表现出来的转接现象不同；反之，同一辅音对不同元音的影响也是不同的。共振峰的转接现象比较复杂，至今尚没找到普遍的规律，但是通过大量的实验人们也发现了一些基本规则。Delattre 在语音合成实验中发现，转接对于辅音的感知十分重要，尤其是后接元音第二共振峰的转接走向与程度，对前面辅音的听辨起着决定性的作用，如果没有这一段转接特征，听起来就不像这个辅音了。他们分析了三个塞音[b,d,g]后接不同的元音[i,e,ɛ,a,ɔ,o,u]时共振峰转接现象(参见图 9.9)，发现尽管不同元音转接的走向与程度是不同的，但同一个辅音造成的共振峰转接走向往往趋于同一点。例如[b]使后接元音的共振峰走向趋于 700Hz 这一点，我们把这一点称为“音轨”。事实上，音轨是由观察到的共振峰转接频率轨迹向前外推 50Hz 而得到的，它表征了辅元转接中共振峰移动起始频率。[d]的音轨在 1800Hz 左右；[g]的情况则不同，它有两个音轨，一个在 3000Hz 左右、另一个在 1200Hz。

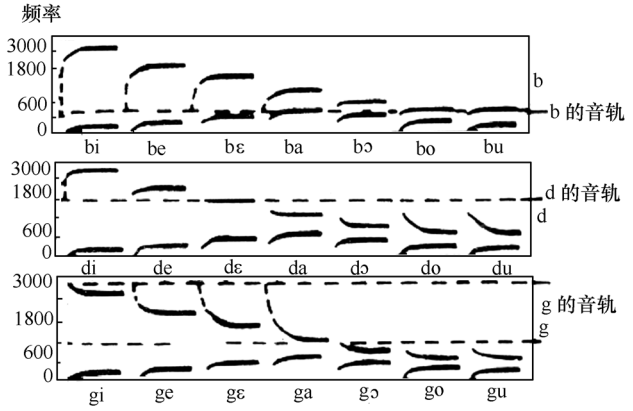


图 9.9　共振峰的转接

对汉语所做的听辨试验也表明：①转接现象主要出现在第二共振峰上，第一与第三共振峰的转接规律比较简单：一般第一共振峰的辅元转接总是向下，音轨为 0Hz，第三共振峰的转接可以忽略；②辅－元转接对辅音听辨的影响，以塞音最大、塞擦音次之、擦音最小；鼻音和边音因为具有元音性质，可不予考虑；③转接音轨与辅音发音位置有密切关系，对照辅音音素表，从左到右，基本上符合音轨逐渐由小变大的原则，但对舌根音[g,k,h]来说，由于它们的发音部位与元音的舌位非常接近，因此它们的音轨与后接元音有关，通常有两个音轨。

由此可见，辅音的发音部位不同时，音轨也就不同。元音舌位也会对音轨产生影响，后高元音[u]对辅音音轨影响最大。

应该指出，音轨本身是从大量实验中得到的统计结果，目前还无法对它作做量分析；但它同下面讲述的元音目标值结合，可以较好地反映辅－元转接规则。

下面我们再看元音之间的音渡问题。在汉语中有 13 个复元音韵母，它们是由两个以上音

素组成。习惯上常把复韵母分为头音(韵头),主元音(韵腹)和尾音(韵尾)三个部分,而且往往把它们看成是几个相对独立和相对稳定的元音。其实不然,复合韵母是一大串飞速滑动过去的音素组合,这种滑动的过程就称为为音渡或者动程。在复合元音的发音过程中,发音器官都处于不断地连续变化之中。例如,发前响二合元音[ai,ao,ou,ei]时,舌位由低到高,口开度也随之由大变小。发后响二合元音[ia,ua,ie,ue,uo]时,则正好相反,舌位由高变低,口开度由小变大。发三合元音[iao,iou]或[uai,uei]时,舌位由高变低又变高,口开度由小变大又变小等。这些反映在复合元音频谱中共振峰是连续变化的,很难确切地划分各个元音之间的界限。图 9.10给出了几个复合元音的声学音渡图,由图可清楚地看出,复合元音中共振峰连续变化的动向。但我们也看到在复合元音的滑动变化过程中出现几个极点(二合元音有两个极点、三合元音有三个极点)。通常所说的头音、主元音和尾音,就是指这些渐变的极点,这些极点称之为元音滑动的目标值。复合元音中的目标值和单个元音情况不同,实验表明:复合元音起始极点的目标值要受前面的邻接辅音影响,一般达不到零声母时的极点位置;主元音的极点位置主要受后接尾音的影响,等等。知道了复合元音极点位置之后,可以用内插的方法得到复合元音的近似共振峰动态轨迹,假如元音滑动轨迹呈现二次曲线特性,那么也可以采用抛物插值方法。一般地说,前响二合元音的共振峰动态轨迹近似线性变化,后响二合元音的共振峰动态轨迹近似曲线,且起始弯曲厉害,后部比较平坦,三合元音的共振峰变化比较复杂,可近似看成二个二合元音。总之,适当选取极点的个数和位置,就可以在一定的范围内改变复合元音的动程和共振峰动态轨迹;运用极点值加内插的方法可以描述汉语韵母内的音渡现象;而音轨至元音目标值的内插可以描述汉语声韵母的转接现象,因此,这样建立起来的共振峰模型对汉语合成有着重要的意义。

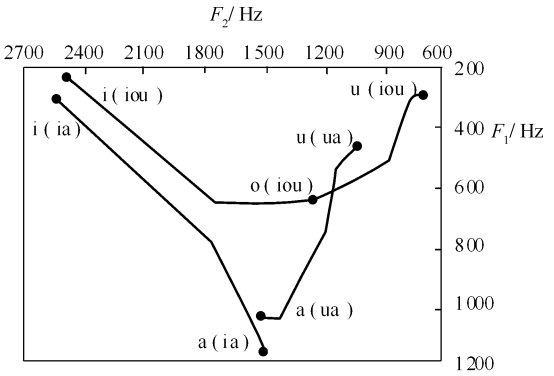


图 9.10　几个复合元音的声学音渡图

需要注意的是:复合元音的共振峰动态轨迹,不可能完全从音渡图上反映出来,因为音渡图只指明了变化的路径,更重要的是要知道变化的速率,也就是共振峰轨迹与时间的对应关系,这将牵涉到复合元音的音长以及各元音的音长比例,在辅元转接中也存在着同样的问题。

汉语中还有 16 个复鼻音尾韵母,它们也都是二个到三个音素组成,尾音是鼻韵尾－n 或－ng,它们和元音复合之后成为一个整体。发音时,发声器官由元音的发音状态逐渐向鼻音的发音状态滑动,最后完全变成鼻音,但这时声带仍然振动,鼻腔没有阻塞,因此鼻韵尾－n 和－ng 具有元音的性质,在建立共振峰动态轨迹时可以近似把它们当做元音一样看待。实验表明,这样近似是可行的,能够反映出鼻韵尾的效果。

3. 声调与变调规则

汉语是一种“声调语音”，在用汉语相互交谈中，人们不但凭不同的声母、韵母（或元音，辅音）来辨别字和词的意义，还需要从不同的声调来区别它们，这就是“声调语音”的特点。例如，星（xing），形（xing），醒（xing），姓（xing）这四个字的音中，声母和韵母都是相同的，但意义不同，这正是声调不同所致。再如：树木，书目；北京，背景；中药，重要等的区别，也是靠声调来实现的。因此汉语的声调具有辨义的功能，它和辅音、元音在语音的区别特征上同样重要。

声调就是音节的高低升降曲折变化，汉语音节的声调主要体现在信号的基音频率随时间而变化的规律上。声调的调值用音高或基音的变化来描写。就不同人来说，妇女和儿童的声音高一些；老年和男人的低一些。同一个人的音高也会有不同，兴奋时的声音略高升，情绪低落时声音略低沉。绝对音高对区别词义是没有作用的，真正对辨义起作用的是音高的相对幅度变化。一般可以从声调的调类、调值和调型来考虑声调特征。对于汉语普通话，声调的调类可以分为阴平、阳平、上声、去声。此外还有一个“轻声”，它是声调的变体。而声调的调值就是声调的实际读法。在传统汉语语音学中，用五度标记法具体描写调值变化，分别用 1、2、3、4、5 表示声调的低、半低、中、半高、高五度。如普通话的 4 个声调的调值：阴平 55 、阳平 35 、上声 214 、去声 51 。调型的简单标记方法是用符号“_、/、∨、＼”冠于音节之上。声调的调型就是从声调起始点高度向右延伸，到达声调结束点的高度连接起来；若是曲线形的声调，就要在转折处再加上一个点，然后把这 3 个点连起来，这就得出了不同的声调调型。根据语音实验，普通话 4 个声调的音高变化如图 9.11 所示。这与基音时变曲线的变化趋势基本相同，因此，一般说可以用基音频率的时变规律来表示声调的变化。

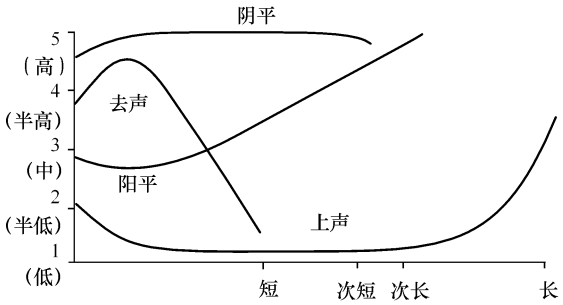


图 9.11　汉语普通话单字调实测值

除了单字调外，还有在音节与音节连续发音时，受语音规律等制约而出现的连续变调。显然连续变调在语句中不应该造成单字调和连续调型太大的变化，否则会产生辩义混淆，甚至误解。这就是说字调和连续变调应该有它们的约定关系和一定模式，在语句中应遵守一定的规律变化，这样纵然语句形式有千变万化，我们仍能听得明白。也就是说，除了音质以外，基本调型还在起着作用，掌握好基本调型以及动态语流中的变调规律，将有助于提高合成汉语的语音质量。

汉语普通话语句中的变调以二字连续变调最为重要，因为二字词在整个汉语词汇中约占 74.3%。当二个字连在一起读时，不论它们是一个词或是一个意群，都会造成变调，其调型原则上是二个字的原单字调型的接续，但受连读的影响会出现变调。变调常常是由后一个字的声调的影响引起，这就是所谓的“逆变规律”。二字调变化规律大致有下列几条：

① 上声字加阴平、阳平、去声、轻声字时，前面的上声字的声调变成半上声。设上声的调值为 214，则半上声的调值为 21，因去掉了调值的上升部分所以叫做半上声。例如，“语音”、

“满意”、“水平”等。

② 两个上声连读，前一个上声变得像阳平，调值由 214 变为 35。如，“五五”、“总理”、“古老”等。

③ 两个去声字相连，前一个去声变成半去，去声字在单独念时是个全降调，从最高的 5 度降到最低的 1 度；而半去则从最高 5 度降到中间值 3 度，即调值为 53。例如“字调”，“论证”，“预报”。

④ 叠字形容词变调，二字重叠做形容词时，第二个字变读阴平。例如，“好好看”，“慢慢走”等，这是顺变规律，可算是规则中的一种特例。

根据上面的变化规律，总结出 16 种二字连续基本调型，如图 9.12 所示，因为上声与上声连读时，前一个上声变成阳平，与阳平上声连读时相同，因此实际上只有 15 种双字调型。

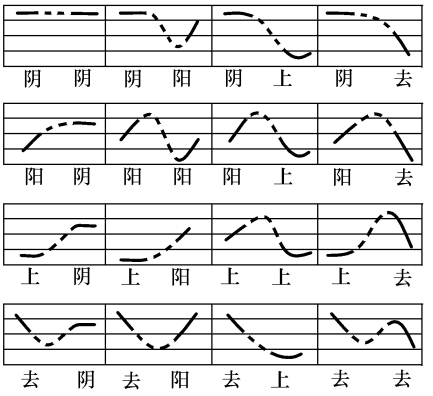


图 9.12　二字连续基本调型

对汉语二字调所做的统计实验，二字调的起始和结尾部分总存在“弯头”和“降尾”的过渡状态，它们约占全声调过程的 10％～15％。二字词第二字的调长比第一个字的调长稍短，大约为第一字调长的 66％。注意到变调规律的这些细节，将会有助于改进合成语音的自然度。有些字在句子中的变调现象比较特殊，例如：不(bu)字，一般情况下念原声调去声；但在后接去声连读时，变调为阳平；“不”字夹在重叠动词或其他词语之间时读轻声，例如，“不!”，“不安”，“不是”，“不可”，“是不是”，“拿不动”，“了不起”等。再如，一(yi)字，单念或在词语末尾时，念本调阳平、去声前念阴平、在其他非去声前念去声、夹在叠用动词之间念轻声。如“一”、“天下第一”、“一生”、“一年”、“一定”、“等一等”。其他尚有一些特殊情况，不一一例举，可参阅有关汉语语音学方面的文献。

在一定的语音环境下，有的音节失去了它原有的声调，念成了一种又轻又短的音，这就是轻声。有时也把轻声看做一种特殊声调变化，即认为汉语有五种声调。轻声一般出现在二字组的后一个字，例如“头”这个字，在“头脑”、“头发”这些词里或单独使用时读本调阳平；在“木头”、“甜头”则读成轻声。轻声音节去调后的音高随前一个字的调型而定。轻声的音高分为高轻、中轻和低轻三种。粗略地说，阴平和阳平后面念中轻、去声后面念低轻；前面一个字如果是上声，一般情况上声加轻声变调为半上加高轻，但如果轻声是由上声字变过来的，除了上面那种变法外，还可以变成阳平加轻声。如：“书上”、“窗户”、“椅子”等。三字组以上的连续变调，一般都可以认为是单音和双字的组合，即使在意义上不完全是这样，但在说话中有说成双音的习惯。三字组在意群上可以有“单双”、“双单”、“单单单”三种形式，例如：“总理讲”三个字由上声字组成，由于“总理”是二字组，它们将按二字变调规律变化：“总”读阳平；“理”虽和后面的

“讲”相邻，且都是上声，但“总理讲”是“双单”格，所以“理”不随“讲”变调。由于说话习惯往往把“单单单”读成“双单”的调型。四字组以“双双”结构的成语居多，五字组以上的情况基本单元仍是单字调和双字调。这对于按规律合成汉语是很有利的。

4. 音长问题

音长也是语音的重要特征之一，对语音的可懂度、自然度都有一定的影响。汉语中音长主要体现在韵母的调型段长度上，调长和调型是密切相关的，通常认为，上声音节最长，阴平、阳平次之，去声最短。在连续语流中调长的变化和声调一样，也要受到连读时上下文的牵连。例如，轻声音节的调长往往比重读时缩短近一半；在双音节中，后一音节的调长要比前一个音节的调长稍短等。在按规则汉语合成中，可将调长和调型一致起来，即凡是平调、升调的调长适中，凡是降升调的调长较长，凡是降调的调长较短，轻声调长最短。声母的音长相对比较稳定。此外，根据实验语音学提供的经验，句子的最后一个音节的调长应比通常情况加长 20％左右。除音长外，音节之间的间隙也对合成语音效果有一定的影响，适当的间隙会使语言听起来更为生动。

第10章　语音识别

10.1　概　述

语音识别以语音为研究对象，它是语音信号处理的一个重要研究方向，是模式识别的一个分支，涉及到生理学、心理学、语言学、计算机科学，以及信号处理等诸多领域，其最终目的是实现人与机器进行自然语言通信，用语言操纵计算机。

语音识别系统的分类方式及依据是根据对说话人说话方式的要求，可以分为孤立字（词）语音识别系统、连接字语音识别系统以及连续语音识别系统。进一步分为两个方向：一是根据对说话人的依赖程度可以分为特定人和非特定人语音识别系统；二是根据词汇量大小，可以分为小词汇量、中等词汇量、大词汇量，以及无限词汇量语音识别系统。

不同的语音识别系统，尽管设计和实现的细节不同，但所采用的基本技术是相似的。一个典型的语音识别系统如图 10.1 所示。主要包括预处理、特征提取和训练识别网络。

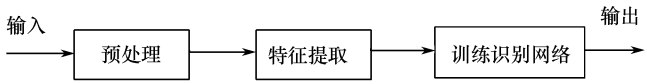


图 10.1　语音识别系统组成部分

10.1.1　预处理

在信号处理系统里，对原始信号进行预处理是必要的，这样可以保证系统获得一个比较理想的处理对象。在语音识别系统中，语音信号的预处理主要包括抗混叠滤波、预加重及端点检测等内容。

1. 抗混叠滤波与预加重

研究表明，语音信号的频谱分量主要集中在 300～3400Hz 的范围内。因此需用一个防混叠的带通滤波器将此范围内的语音信号的频谱分量取出，然后对语音信号进行采样，得到离散的时域语音信号。根据采样定理，如果模拟信号的频谱的带宽是有限的（例如，不包含高于 f_m 的频率成分），那么用等于或高于 $2f_m$ 的取样频率进行采样，则所得到的信号能够完全唯一的代表原模拟信号，或者说能够由取样信号恢复出原始信号。实际应用中，大多数情况选用 8kHz 的采样频率。尽管如此，还必须顾及到语音信号本身包含着 4kHz 以上频率成分这样一个事实。即使有的语音的频谱能量主要集中在低频段，但由于噪声环境的宽带随机噪声叠加的结果，使得在采样之前，语音信号总包含着 4kHz 以上的频率成分。因此，为了防止混叠失真和噪声干扰，必须在采样前用一个锐截止模拟低通滤波器对语音信号进行滤波。该滤波器称为反混叠滤波器或去伪滤波器。

语音从嘴唇辐射会有 6dB/oct 的衰减，因此在对语音信号进行处理之前，希望能按 6dB/oct的比例对信号加以提升（或加重），以使得输出信号的电平相近似。当用数字电路来实现 6dB/oct 预加重时，可采用以下差分方程所定义的数字滤波器：

$$y(n)=x(n)-ax(n-1) \tag{10.1}$$

式中，系数 a 常在 0.9～1 之间选取。

2. 端点检测

语音信号起止点的判别是任何一个语音识别系统必不可少的组成部分。因为只有准确地找出语音段的起始点和终止点，才有可能使采集到的数据是真正要分析的语音信号，这样不但减少了数据量、运算量和处理时间，同时也有利于系统识别率的改善。常用的端点检测方法有下面两种。

(1) 短时平均幅度

端点检测中需要计算信号的短时能量，由于短时能量的计算涉及到平方运算，而平方运算势必扩大了振幅不等的任何相邻取样值之间的幅度差别，这就给窗的宽度选择带来了困难，因为必须用较宽的窗才能对取样间的平方幅度起伏有较好的平滑效果，然而又可能导致短时能量反映不出语音能量的时变特点。而用短时平均幅度来表示语音能量，在一定程度上可以克服这个弊端。

(2) 短时平均过零率

当离散信号的相邻两个取样值具有不同的符号时，便出现过零现象，单位时间内过零的次数叫做过零率。如果离散时间信号的包络是窄带信号，那么过零率可以比较准确地反映该信号的频率。在宽带信号情况下，过零率只能粗略的反映信号的频谱特性。

在前面的第 3 章中，介绍了使用两级判决法的端点检测技术，可参考。

10.1.2 语音识别特征提取

语音识别的一个重要步骤是特征提取，有时也称为前端处理，与之相关的内容则是特征间的距离度量。所谓特征提取，即对不同的语音寻找其内在特征，由此来判别出未知语音，所以每个语音识别系统都必须进行特征提取。特征的选择对识别效果至关重要，选择的标准应体现对异音字之间的距离尽可能大，而同音字之间的距离应尽可能小。若以前者距离与后者距离之比为优化准则确定目标量，则应是该量最大。同时，还要考虑特征参数的计算量，应在保持高识别率的情况下，尽可能减少特征维数，以减小存储要求和利于实时实现。

孤立词语音识别系统的特征提取一般需要解决两个问题，一个是从语音信号中提取(或测量)有代表性的合适的特征参数(即选取有用的信号表示)；另一个是进行适当的数据压缩。而对于非特定人语音识别来讲，则希望特征参数尽可能多地反映语义信息，尽量减少说话人的个人信息(对特定人语音识别来讲，则相反)。从信息论角度讲，这也是信息压缩的过程。

语音信号的特征主要有时域和频域两种。时域特征如短时平均能量、短时平均过零率、共振峰、基音周期等；频域特征有线性预测系数(LPC)、LP 倒谱系数(LPCC)、线谱对参数(LSP)、短时频谱、Mel 频率倒谱系数(MFCC)等。现在还有结合时间和频率的特征，即时频谱，充分利用了语音信号的时序信息。基于听觉模型的特征参数提取，如感知线性预测(PLP)分析，试图从不同于声道模型的另一个方面进行研究。所有这些特征都只包含了语音信号的部分信息。为了充分表征语音信号，人们尝试综合各种特征，并取得了一定的效果。但由于目前语音识别分类器的限制和数学模型描述的局限性，人们尚未充分利用已有的部分信息，于是特征的变换与取舍、特征时序信息的使用等成了重要的研究课题。有关特征研究的另外一个重要方面是特征的抗噪声性能，由于语音识别的最终目标是在现实世界中使用，背景噪声的干扰成为不可忽视的因素，因此必须研究一种方法，使得特征的提取尽可能不受噪声的影响。下

面介绍几种特征提取方法。

1. 线性预测系数(LPC)

线性预测分析从人的发声机理入手,通过对声道的短管级联模型的研究,认为系统的传递函数符合全极点数字滤波器的形式,从而某一时刻的信号可以用前若干时刻的信号的线性组合来估计。通过使实际语音的采样值和线性预测采样值之间达到最小均方误差(MSE),即可得到线性预测系数 LPC。

根据语音产生的模型,语音信号 $S(z)$是一个线性非移变因果稳定系统 $V(z)$受到信号 $E(z)$激励产生的输出。在时域中,语音信号 $s(n)$是该系统的单位取样响应 $v(n)$和激励信号 $e(n)$的卷积。语音产生的声道模型在大多数情况下是一个可用式(10.2)阐述的全极点模型:

$$H(z)=\frac{1}{1-\sum_{k=1}^{p}a_k z^{-k}} \tag{10.2}$$

根据最小均方误差对该模型参数 a_k 进行估计,就得到了线性预测编码(LPC)算法,求得的 $\hat{a}_p$ 即为 LP 系数(p 为预测器阶数)。对 LPC 的计算方法有自相关法(Levinson-Durbin)、协方差法、格型法等。计算上的快速有效保证了这一声学特征的广泛使用。

2. LPC 倒谱系数(LPCC)

倒谱系数是信号的 z 变换的对数模函数的逆 z 变换,一般先求信号的傅里叶变换,取模的对数,再求傅里叶逆变换得到。既然线性预测也是一种参数谱估计方法,而且其系统函数的频率响应 $H(e^{j\omega})$反映了声道的频率响应和被分析信号的谱包络,因此用 $\log|H(e^{j\omega})|$做傅里叶逆变换求出的倒谱系数,应该是一种描述信号的良好参数。其主要优点是比较彻底地去掉了语音产生过程中的激励信息,反映了声道响应,而且往往只需要几个倒谱系数就能够很好地描述语音的共振峰特性。

3. Mel 频率倒谱系数(MFCC)

Mel 频率倒谱系数是先将信号频谱的频率轴转变为 Mel 刻度,再变换到倒谱域得到倒谱系数。其计算过程如下:

① 将信号进行短时傅里叶变换得到其频谱。

② 求频谱幅度的平方,即能量谱,并用一组三角滤波器在频域对能量进行带通滤波。这组带通滤波器的中心频率是按 Mel 频率刻度均匀排列的(间隔 150Mel,带宽 300Mel),每个三角滤波器的中心频率的两个底点的频率分别等于相邻的两个滤波器的中心频率,即每两个相邻的滤波器的过渡带互相搭接,且频率响应之和为 1。滤波器的个数通常与临界带数相近,设滤波器数为 M,滤波后得到的输出为:$X(k)$, $k=1,2,\cdots,M$。

③ 对滤波器的输出取对数,然后做 $2M$ 点傅里叶逆变换即可得到 MFCC。由于对称性,此变换可简化为:

$$C_n=\sum_{k=1}^{M}\log X(k)\cos[\pi(k-0.5)n/M],\quad n=1,2,\cdots,L \tag{10.3}$$

这里,MFCC 系数的个数 L 通常取最低的 12～16。在谱失真测度定义中通常不用 0 阶倒谱系数,因为它是反映倒谱能量的。上面所说的在频域进行带通滤波是对能量谱进行滤波,这样做的根据是考虑到一个多分量信号的总能量应该是各个正交分量的能量之和。

4. 过零峰值幅度(ZCPA)

特征参数的好坏直接决定着系统的识别性能。要想使识别系统有好的鲁棒性,必须要求提取的特征参数有很强的抗噪性。经典的特征参数在无噪声环境下都取得了相当好的效果,但在噪声环境下,系统的识别率会显著下降。人类的听觉系统在噪声环境下能够很好工作,所以如果语音识别系统能模拟人类听觉感知的处理特点,噪声环境下识别率一定会提高。近年来,基于听觉模型的语音特征提取方法在语音识别领域日益受到重视,就是因为听觉模型最接近人耳对声音信号的处理过程,提取的特征能反映声音的本质,具有很好的鲁棒性。过零峰值幅度特征 ZCPA 就是基于人类听觉特性的一种特征。

人耳由外耳、中耳、内耳三部分构成。语音信号在外耳的耳膜上转化为机械振动,通过中耳传递到内耳的耳蜗上,中耳充当外耳和内耳的匹配阻抗。而语音信号的主要处理任务是在内耳中进行的,尤其是在内耳的耳蜗中进行的。耳蜗中的基底膜对外来的声音信号有频率选择和调谐的作用,在耳蜗基部通过前庭窗传递来的语音信号被转化为基底膜的行波,沿基底膜传播,其峰值出现在基底膜的不同位置。频率越低,振动峰值位置越靠近蜗孔,随频率增高,该峰值越靠近基底膜根部。约 800Hz 以上,声音频率沿基底膜按对数分布。其位移和频率的关系为

$$F=A(10^{ax}-1) \tag{10.4}$$

其中,F 是频率(Hz),x 是基底膜的归一化距离,A 和 a 是常数,分别为 $A=165.4$、$a=2.1$。

在听觉系统中耳蜗对声音的感受和换能作用是整个复杂的听觉系统中非常重要的一个环节,同时耳蜗具有串/并转换器的功能,它实际上相当于一组并联的带通滤波器,串行输入的声音信号在耳蜗中被分解并以多路并行的方式输出。这样为仿真耳蜗滤波器的模型提供了一定的依据。图 10.2 给出了基于人耳听觉特性的 ZCPA 特征提取原理图。

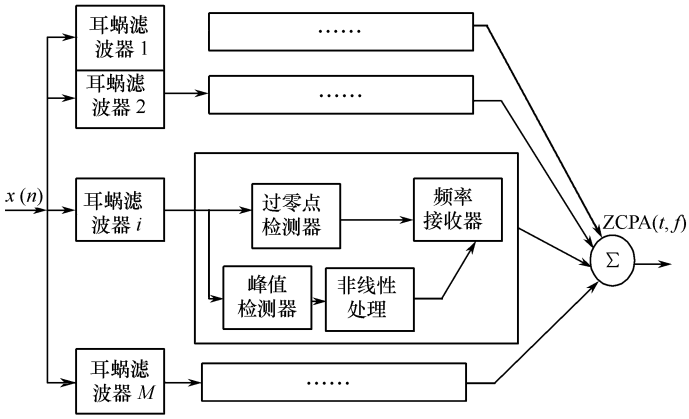


图 10.2　ZCPA 系统原理框图

该系统由带通滤波器组、过零检测器、峰值检测器、非线性压缩和频率接收器组成。带通滤波器组由 16 个 FIR 滤波器组成,用来仿真耳蜗基底膜;过零检测器、峰值检测器、非线性压缩部分则仿真听觉神经纤维。从过零检测器获得频率信息,峰值检测器获得强度信息,经非线性压缩后,用频率接收器合成频率信息和强度信息,最后将 16 路所获得的信息合成为语音信号的特征。

分析表明:在噪声存在的情况下,随着门限值的提高,门限跨越的间隔扰动也变得越大,此时过零率就显得更具有鲁棒性,因此它能够提供一种较好的用于噪声环境下的语音信号表示

方法。ZCPA 模型的原理与传统的信号处理方案有显著的不同，它需要测量信号在一个时间段内的瞬时频率和强度信息，并在随后需要进行一个时域信息的积累操作以获取最终输出。

10.1.3 语音识别方法

一般来说，语音识别的方法有四种：基于声道模型和语音知识的方法、模式匹配的方法、统计模型方法以及利用人工神经网络的方法。基于声道模型和语音知识的方法起步较早，在语音识别技术提出的开始，就有了这方面的研究，但由于其模型及语音知识过于复杂，现阶段没有达到实用的阶段。目前常用的方法是后三种方法，目前它们都已达到了实用阶段。模式匹配常用的技术有矢量量化(VQ)和动态时间规整(DTW)；统计型模型方法常见的是隐马尔可夫模型(HMM)；语音识别常用的神经网络有反向传播(BP)网络、径向基函数网络(RBF)及小波网络。本书重点介绍经典的隐马尔可夫模型及其在语音识别中的应用。

模式匹配法用于语音识别共有四个步骤：特征提取、模板训练、模板分类、判决。图 10.3 是模式匹配法的原理框图。

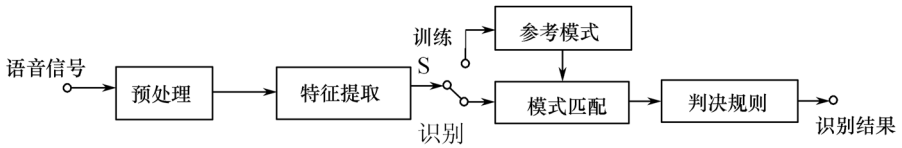


图 10.3 语音识别系统模式匹配法原理框图

在该图中，语音经过话筒变成电信号(即图中语音信号)后加在识别系统输入端。经过预处理后，语音信号的特征被提取出来，首先在此基础上建立所需的模板，这个建立模板的过程称为训练过程。接下来将新提取的特征与模板匹配的过程称为识别过程。即根据语音识别的整体模型，将输入的语音信号的特征与已经存在的语音模板(参考模式)进行比较，根据一定的搜索和匹配策略(判决规则)，找出一系列最优的与输入的语音相匹配的模板。然后，根据此模板号的定义，通过查表就可以给出计算机的识别结果。

由于在训练或识别过程中，即使同一个人发同一个音时，不仅其持续时间长度会随机地改变，而且各音素的相对时长也是随机变化的。因此在匹配时如果只对特征向量系列进行线性时间规整，其中的音素就有可能对不准。20 世纪 60 年代日本学者板仓(Itakura)提出了动态时间规整(DTW)算法。算法的思想就是把未知量均匀地伸长或缩短，直到它与参考模式的长度一致时为止。在时间规整过程中，未知单词的时间轴要不均匀地扭曲或弯折，以便使其特征与模型特征对正。

DTW 是较早的一种模式匹配和模型训练技术，它应用动态规划方法成功解决了语音信号特征参数序列比较时时长不等的难题，在孤立词语音识别中获得了良好性能。但因其不适合连续语音大词汇量语音识别系统，目前已被 HMM 模型和 ANN 替代。

隐马尔可夫模型是对语音信号的时间序列结构建立统计模型，将之看做一个数学上的双重随机过程：一个是用具有有限状态数的马尔可夫链来模拟语音信号统计特性变化的隐含的随机过程，另一个是与马尔可夫链的每一个状态相关联的观测序列的随机过程。前者通过后者表现出来，但前者的具体参数是不可测的。人的言语过程实际上就是一个双重随机过程，语音信号本身是一个可观测的时变序列，是由大脑根据语法知识和言语需要(不可观测的状态)发出的音素的参数流。可见，HMM 合理地模仿了这一过程，很好地描述了语音信号的整体非

平稳性和局部平稳性，是较为理想的一种语音模型。

与模式匹配法相比，HMM 是一种迥然不同的概念。在模式匹配法中，“参考样本”是由事先存储起来的“模式”本身充当的，而 HMM 则是把这一“参考样本”用一个数字模型来表示（马尔可夫链），然后待识别的语音与这一数学模型相比较，这就从概念上较前深化了一步。图 10.4给出了一个基于 HMM 的孤立词语音识别原理图。

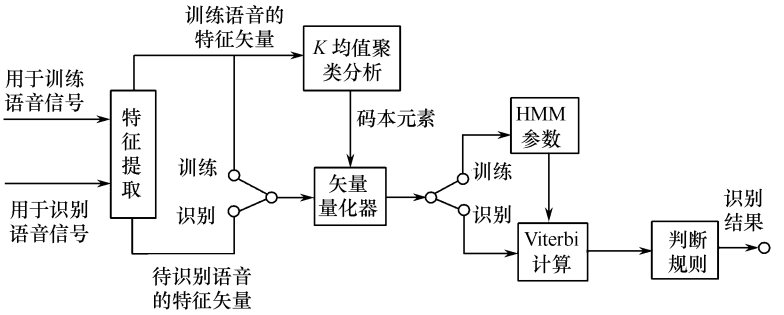


图 10.4　隐马尔可夫模型用于孤立词语音识别框图

采用 HMM 进行语音识别，实质上是一种概率运算。根据训练集数据计算得出模型参数后，测试集数据只需分别计算各模型的条件概率（Viterbi 算法），取此概率最大者即为识别结果。由于马尔可夫过程各状态间的转移概率和每个状态下的输出都是随机的，故这种模型更能适应语音发音的各种微妙的变化，使用起来比模板匹配方法灵活得多。除训练时需运算量较大外，识别时的运算量仅有模式匹配法的几分之一。

人工神经网络（ANN）在语音识别中的应用是当前研究的热点。人工神经网络本质上是一个自适应非线性动力学系统，模拟了人类神经元活动的原理，具有自适应性、并行性、鲁棒性、容错性和学习特性。目前用于语音识别的神经网络有多层感知机，Kohonen 自组织神经网和预测神经网。

人工神经网络是采用物理上可实现的系统来模拟人脑神经细胞的结构和功能的系统。它是由很多简单的处理单元有机地连接起来进行并行的工作，人工神经网络中大量神经元并行分布运算的原理、高效的学习算法以及对人的认知系统的模仿能力等都使它极适宜于解决类似于语音识别这一类课题。由于神经网络反映了人脑功能的基本特征，具有自组织性、自适应性和连续学习的能力。这种网络是可以训练的，即可以随着经验的积累而改变自身的性能。同时由于高度的并行性，它们能够进行快速判决并具有容错性，特别适合于解决像语音识别这类难以用算法来描述而又有大量样本可供学习的问题，图 10.5 给出了神经网络用于语音识别的原理图。

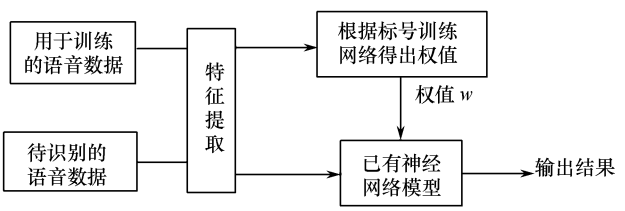


图 10.5　基于神经网络的语音识别原理图

神经网络的一项非常重要的功能是通过学习实现对于输入矢量的分类。这就是说每输入一个矢量，人工神经网络输出一个该矢量所属类别的标号。在传统的语音识别方法中，通过特

征参数的提取及模式匹配完成识别。由于语音信号的高度多变性，输入模式要与标准模式完全匹配是几乎不可能的。神经网络的语音识别方法与传统方法的差异在于提取了语音的特征参数后，不像传统方法那样有输入模式与标准模式的比较匹配及统计参数，而是靠神经网络中大量的连接权对输入模式进行非线性运算，产生最大兴奋的输入点就代表了输入模式对应的分类。神经网络的连接权系数是在使用中根据识别结果的正确与否不断地进行自适应修正。比较起来，神经网络识别系统更接近人类的感知过程。

矢量量化(VQ)技术是20世纪70年代后期发展起来的一种数据压缩和编码技术，广泛应用于语音编码、语音合成、语音识别和说话人识别等领域。矢量量化在语音信号处理中占有十分重要的地位，在许多重要的研究课题中，矢量量化都起着非常重要的作用。

矢量量化技术在语音识别中应用时，一般是先用矢量量化的码本作为语音识别的参考模板，即系统词库中的每一个(字)词，做一个码本作为该(字)词的参考模板。识别时对于任意输入的语音特征矢量序列 $X_1, X_2, \cdots, X_N$，计算该序列对每一个码本的总平均的失真量化误差，即语音每一帧特征矢量与码本的失真之和除以该语音的长度(帧数)。总平均失真误差最小的码本所对应的(字)词即为识别结果，这一过程如图10.6所示。

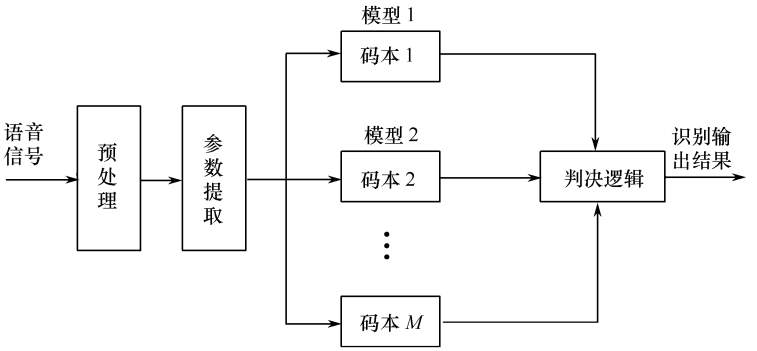


图10.6　矢量量化在语音识别中的应用

10.2　HMM基本原理及在语音识别中的应用

10.2.1　隐马尔可夫模型

马尔可夫过程(或马尔可夫链)描述的是一类重要的随机过程，它是由俄国数学家A. Markov于1907年提出来的。它的直观解释是：在已知系统目前的状态(现在)的条件下，“将来”与“过去”无关。这种过程也称为无记忆的单随机过程。如果这种单随机过程的取值(状态)是离散的，我们又可以将它称为无记忆的离散随机过程。假设有一个系统，它在任何时间可以认为处在有限多个状态的某个状态下。在均匀划分的时间间隔上，系统的状态按一组概率发生改变(包括停留在原状态)，这组概率值和状态有关，而且这个状态对应于一个可观测的物理事件，因此称为可观测马尔可夫过程。相对马尔可夫过程，人们又提出了一种状态及其行为都为不可测(随机)的双随机过程。从外界来看，这种过程的状态是随机且不可见(隐藏)的，这是一个基本随机过程。这个过程只能通过另一组随机过程才能观测到，另一组随机过程产生出观测序列(行为)，而这组行为是可见不可测的。因此，这种双随机过程称为隐马尔可夫

模型(或隐马尔可夫过程)。通常,HMM 对应的状态被假设为离散的,且其演变是无记忆的,因而,HMM 也被称为无记忆的离散双随机过程。

隐马尔可夫过程是一个双重随机过程:一重用于描述非平稳信号的短时平稳段的统计特征(信号的瞬态特征,可直接观测到);另一重随机过程描述了每个短时平稳段如何转变到下一个短时平稳段,即短时统计特征的动态特性(隐含在观察序列中)。基于这两重随机过程,HMM 既可有效解决怎样辨识具有不同参数的短时平稳信号段,又可解决怎样跟踪它们之间的转化等问题。

人的言语过程也是这样一个双重随机过程。因为语音信号本身是一个可观察的序列,而它又是由大脑里的(不可观察的)、根据言语需要和语法知识(状态选择)所发出的音素(词、句)的参数流,同时,大量实验表明,HMM 的确可以非常精确地描述语音信号的产生过程。

一个隐马尔可夫模型由下列参数来决定:

① N——模型的状态数目。虽然 HMM 的状态是隐藏起来的,但在许多实际应用中,模型的状态常常有某种物理意义,状态的集合表示为 $S=\{S_1,S_2,\cdots,S_N\}$,t 时刻的状态表示为 q_t。

② M——观测符号数。即每个状态可能输出的观测符号的数目。观测符号集合表示为 $O=\{O_1,O_2,\cdots,O_M\}$。

③ A——状态转移概率分布。这是由状态转移概率构成的矩阵。

$$A=\{a_{ij}\},a_{ij}=P[q_{t+1}=S_j\mid q_t=S_i],1\leqslant i,j\leqslant N \tag{10.5}$$

其中 a_{ij} 具有以下性质:

$$a_{ij}\geqslant 0 \quad \text{且}\sum_{j=1}^{N}a_{ij}=1 \tag{10.6}$$

④ B——状态 S_j 的观测符号概率分布。

$$B=\{b_j(O_k)\},b_j(O_k)=P[\text{在 } t \text{ 时刻的输出符号为 } O_k\mid q_t=S_j]$$

$$1\leqslant j\leqslant N,1\leqslant k\leqslant M \tag{10.7}$$

⑤ π——初始状态分布。

$$\pi=\{\pi_i\},\pi_i=P[q_1=S_i],1\leqslant i\leqslant N \tag{10.8}$$

为了完整地描述一个隐马尔可夫模型,应当指定状态数 N,观测符号数 M,以及三个概率密度 A、B 和 π。这些参数之间有一定的联系,因此为了方便,HMM 常用 $\lambda=(A,B,\pi)$ 来简记。

10.2.2 隐马尔可夫模型的三个基本问题

给定 HMM 的形式后,为了将其应用于实际,必须解决以下三个基本关键问题:

● 已知观测序列 $O=\{O_1,O_2,\cdots,O_T\}$ 和模型 $\lambda=(A,B,\pi)$,如何有效的计算在给定模型条件下产生观测序列 O 的概率 $P(O|\lambda)$。

● 已知观测序列 $O=\{O_1,O_2,\cdots,O_T\}$ 和模型 $\lambda=(A,B,\pi)$,如何选择相应的在某种意义上最佳的(最好解释观测序列的)状态序列。

● 给定观测序列,如何调整参数(A,B,π)使条件概率 $P(O|\lambda)$最大。

1. 第一个问题的求解

这是一个评估问题,即已知模型和一个观测序列,怎样来评估这个模型(它与给定序列匹

配得如何),或怎样给模型打分,这个问题通常被称为“前向－后向”的算法解决。

(1) 前向算法

首先要定义一个前向变量 $\alpha_t(i)$:

$$\alpha_t(i)=P(O_1,O_2,\cdots,O_t;q_t=S_i|\lambda) \tag{10.9}$$

即在给定模型 λ 的条件下,产生 t 以前的部分观测符号序列(包括 O_t 在内)$\{O_1,O_2,\cdots,O_t\}$,且 t 时刻又处于状态 S_i 的概率。以下是前向变量进行迭代计算的步骤:

① 初始化

$$\alpha_1(i)=\pi_i b_i(O_1),1\leqslant i\leqslant N \tag{10.10}$$

② 迭代计算

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1}),1\leqslant t\leqslant T-1;1\leqslant j\leqslant N \tag{10.11}$$

③ 最后计算

$$P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i) \tag{10.12}$$

其中 a_{ij} 为状态转移矩阵中的元素,$b_j(O_t)$ 为观测符号矩阵中的元素。

(2) 后向算法

同理,可以类似地定义后向变量 $\beta_t(i)$:

$$\beta_t(i)=P(O_{t+1},O_{t+2},\cdots,O_T;q_t=S_i|\lambda) \tag{10.13}$$

即在给定模型 λ 及 t 时刻处于状态 S_i 的条件下,产生 t 以后的部分观测符号序列$\{O_{t+1},O_{t+2},\cdots,O_T\}$的概率。后向变量也可以用迭代法进行计算,步骤如下:

① 初始化

$$\beta_T(i)=1,1\leqslant i\leqslant N \tag{10.14}$$

② 迭代计算

$$\beta_t(i)=\sum_{i=1}^{N}a_{ij}b_j(O_{t+1})\beta_{t+1}(j),t=T-1,T-2,\cdots,1,1\leqslant i\leqslant N \tag{10.15}$$

(3) 最后计算

$$P(O|\lambda)=\sum_{i=1}^{N}\beta_1(i) \tag{10.16}$$

前向和后向算法对于求解第一个问题和第二个问题也是有帮助的。

由于 $\alpha_t(i)$表示 t 时刻处于状态 S_i 且部分观测序列为$\{O_1,O_2,\cdots,O_t\}$,而 $\beta_t(i)$表示 t 时刻处于状态 S_i 且剩下部分的观测序列为$\{O_{t+1},O_{t+2},\cdots,O_T\}$,因而 $\alpha_t(i)$、$\beta_t(i)$表示产生整个观测序列 O 且 t 时刻处于状态 S_i 的概率,即

$$\alpha_t(i)\beta_t(i)=P(O,q_t=S_i|\lambda) \tag{10.17}$$

那么,第一个问题也可以通过同时使用前向后向概率来求解,即

$$P(O|\lambda)=\sum_{i=1}^{N}\alpha_t(i)\beta_t(i),t=1,2,\cdots,T;i=1,2,\cdots,N \tag{10.18}$$

2. 第二个问题的求解

这个问题是求取伴随给定观测序列产生的最佳状态序列。这一最佳判据,目的就是要使

正确的状态数目的期望值最大。它通常用 Viterbi 算法解决，用于模型细调。

为了得到问题二的求解，首先定义变量 $\gamma_t(i)$ 为

$$\gamma_t(i)=P(q_t=S_i|O,\lambda) \tag{10.19}$$

它是在给定观测序列 O 和模型 λ 的条件下，t 时刻处在状态 S_i 的概率，则根据式(10.18)，$\gamma_t(i)$ 可用前后向变量表示为

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}=\frac{\alpha_t(i)\beta_t(i)}{P(O\mid\lambda)} \tag{10.20}$$

根据式(10.17)

$$\gamma_t(i)=\frac{P(O,q_t=S_i|\lambda)}{P(O|\lambda)} \tag{10.21}$$

且有

$$\sum_{i=1}^{N}\gamma_t(i)=1 \tag{10.22}$$

利用 $\gamma_t(i)$，可以求出在各个时刻所处的最可能的状态为

$$q_t=\underset{1\leqslant i\leqslant N}{\operatorname{argmax}}[\gamma_t(i)],1\leqslant t\leqslant T \tag{10.23}$$

但是，上式的求解仅仅从每个时刻出现最可能的状态来考虑的，而没有考虑到状态序列的发生概率(如没有考虑全局结构，时间上相邻状态以及观测序列的长度，等等)。

上述问题的解决办法是对最佳判据进行修正。最广泛应用的判据是寻找单个最佳状态序列(路径)，亦即使 $P(Q|O,\lambda)$ 最大。下面介绍的 Viterbi 算法就是一种以动态规划为基础的寻找单个最佳状态序列的方法。

首先定义一个变量 $\delta_t(i)$

$$\delta_t(i)=\max_{q_1,q_2,\cdots,q_{t-1}}P[q_1,q_2,\cdots,q_t=S_i,O_1,O_2,\cdots,O_t|\lambda] \tag{10.24}$$

即 $\delta_t(i)$ 意为在 t 时刻，沿着一条路径抵达状态 S_i，并生成观察序列 $\{O_1,O_2,\cdots,O_t\}$ 的最大概率。$\delta_t(i)$ 可用迭代法进行计算

$$\delta_{t+1}(j)=[\max_i\delta_t(i)a_{ij}]b_j(O_{t+1}) \tag{10.25}$$

为了实际找到这个状态序列，需要跟踪使式(10.25)最大的参数变化的轨迹(对每个 t 和 j)，即为了能够得到最优的状态序列，在求解过程中，对每一个时刻和状态，需要保留使得上式中最大化条件得以满足的上一刻的状态。可以借助阵列来做到这一点，完整的算法如下所述。

① 初始化

$$\delta_1(i)=\pi_ib_i(O_1),1\leqslant i\leqslant N,\psi_1(i)=0 \tag{10.26}$$

② 迭代计算

$$\delta_t(j)=\max_{1\leqslant i\leqslant N}[\delta_{t-1}(i)a_{ij}]b_j(O_t),2\leqslant t\leqslant T,1\leqslant j\leqslant N \tag{10.27}$$

$$\psi_t(j)=\underset{1\leqslant i\leqslant N}{\operatorname{argmax}}[\delta_{t-1}(i)a_{ij}],2\leqslant t\leqslant T,1\leqslant j\leqslant N \tag{10.28}$$

③ 最后计算

$$p^* = \max_{1\leqslant i\leqslant N}[\delta_T(i)] \tag{10.29}$$

$$q_T^* = \operatorname{argmax}_{1\leqslant i\leqslant N}[\delta_T(i)] \tag{10.30}$$

④ 路径(状态序列)回溯

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \cdots, 1 \tag{10.31}$$

3. 第三个问题的求解

这个问题是调整模型参数(A,B,π),使观测序列在给定模型条件下发生概率最大。即模型参数重估问题(训练问题)。事实上,给定任何有限观测序列作为训练数据,没有一种最佳方法能估计模型参数。但是可以利用迭代处理方法(Baum-Welch 法,或称期望值修正法)来选择(A,B,π)以使得 $P(O|\lambda)$最大,可以用参数重估来解决。

为了说明问题,首先定义变量 $\xi_t(i,j)$

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \tag{10.32}$$

即给定模型和观测序列条件下,在时间 t 处于状态 S_i,而在时间 $t+1$ 处于状态 S_j 的概率。根据前后向变量的定义,从图 10.7 可以看出,$\xi_t(i,j)$可写成如下形式

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O\mid\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{10.33}$$

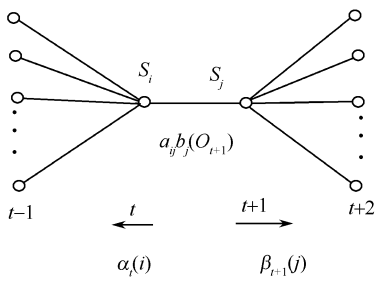


图 10.7　$\xi_t(i,j)$计算示意图

此前已经定义了 $\gamma_t(i)$为在给定模型 λ 和观察序列 O 的条件下,在时刻 t 位于状态为 S_i 的条件概率,将 $\xi_t(i,j)$对 j 求和,可把两者联系起来,即

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j) \tag{10.34}$$

利用上面的公式及计算事件发生的概念,可以得到估计隐马尔可夫模型参数的方法,其计算公式如下[参考式(10.20)和式(10.32)]:

(1) π 的重估公式

$$\bar{\pi}_i = \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum_{i=1}^{N}\alpha_1(i)\beta_1(i)} \tag{10.35}$$

即在时间 $t=1$ 处于状态 S_i 的次数的期望值。

(2) a_{ij} 的重估公式

$$\bar{A}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} = \frac{\sum_{t=1}^{T-1}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)/P(O\mid\lambda)}{\sum_{t=1}^{T-1}\alpha_t(i)\beta_t(i)/P(O\mid\lambda)} \tag{10.36}$$

将 $\gamma_t(i)$ 对 t(t 从 1 到 $T-1$)求和,将得到一个量,可解释为从状态 S_i 进行转移的次数的期望值。类似地,将 $\xi_t(i,j)$ 对时间 t(t 从 1 到 $T-1$)求和,可以得到从状态 S_i 转移到 S_j 的期望值。

即从状态 S_i 转移到状态 S_j 次数的期望与从状态 S_i 转移出去次数的期望的比值。

(3) $b_j(O_k)$ 的重估公式

$$\overline{B}_j(O_k)=\frac{\sum_{t=1}^{T}\gamma_t(j)_{O_k=v_k}}{\sum_{t=1}^{T}\gamma_t(j)}=\frac{\sum_{\substack{t=1\\O_k=v_k}}^{T}\alpha_t(i)\beta_t(i)/P(O\mid\lambda)}{\sum_{t=1}^{T}\alpha_t(i)\beta_t(i)/P(O\mid\lambda)} \tag{10.37}$$

它表示在状态 S_j 观测到符号 v_k 的次数的期望与出现状态 S_j 的次数的期望之比。

把现在的模型定义为 $\lambda=(A,B,\pi)$,把重估模型定义为 $\overline{\lambda}=(\overline{A},\overline{B},\overline{\pi})$。以上述方法为基础,如果不断地用 $\overline{\lambda}$ 代替 λ,并重复上述重估计算,那么就能够改善由模型观测到 O 的概率,直到达到某个极限点为止。

4. 解决下溢问题后的重估公式

我们可以看到上面的重估公式均涉及到了前向变量和后向变量的计算。而每个前向变量 $\alpha_t(i)$ 和后向变量 $\beta_t(j)$ 都是通过递推计算得到的,即是由连续相乘的概率值组成的。当 t 达到较大数值(如 100)时,二者的动态范围会超过任何计算机的精度范围从而导致下溢,因此要用软件实现此算法,必须在计算过程中使用定标算法。即每递推计算一次便对运算结果乘以一个适当放大的比例因子。下面给出了详细的定标过程并且推导了加入定标因子后三个参数的重估公式(包括单序列和多序列重估公式)。

定标的基本方法是对 $\alpha_t(i)$ 和 $\beta_t(j)$ 乘以一个定标系数,该系数与 i 无关(即它只取决于 t),目的是使定标后的 $\alpha_t(i)$ 和 $\beta_t(j)$ 总是处在计算机的动态范围之内,在计算结束后,应当去掉所有的定标系数。下面给出完整的定标过程。

(1) 对前向变量进行定标

定标过程需要引入几个新的变量:$\bar{\alpha}_t(i)$ 和 $\hat{\alpha}_t(i)$。$\alpha_t(i)$ 是待求前向变量值,设 $\bar{\alpha}_t(i)$ 为递推值,$\hat{\alpha}_t(i)$ 为修正递推值,由于 $\alpha_t(i)$ 的下溢问题,在实际计算过程中这个变量不能出现,所以公式中的 $\alpha_t(i)$ 必须用修正递推值 $\hat{\alpha}_t(i)$ 代替。设 c_t 为标度(定标)因子

$$c_t=\frac{1}{\sum_{i=1}^{N}\bar{\alpha}_t(i)} \tag{10.38}$$

则前向变量的递推计算按下面步骤进行。

初始化:

$$\bar{\alpha}_1(i)=\alpha_1(i) \tag{10.39}$$

$$\hat{\alpha}_1(i)=c_1\bar{\alpha}_1(i)=c_1\alpha_1(i) \tag{10.40}$$

$$c_1=\frac{1}{\sum_{i=1}^{N}\bar{\alpha}_1(i)}=\frac{1}{\sum_{i=1}^{N}\alpha_1(i)}=\frac{1}{P(O_1\mid\lambda)} \tag{10.41}$$

递推:

$$\hat{\alpha}_t(i)=c_t\bar{\alpha}_t(i) \tag{10.42}$$

定标后前向变量的计算公式为：

$$\bar{\alpha}_{t+1}(j)=\sum_{i=1}^{N}\hat{\alpha}_t(i)a_{ij}b_j(O_{t+1}) \tag{10.43}$$

根据上两个公式可得

$$\begin{aligned}\hat{\alpha}_2(j)&=c_2\bar{\alpha}_2(j)=c_2\sum_{i=1}^{N}\hat{\alpha}_1(i)a_{ij}b_j(O_2)\\&=c_2c_1\sum_{i=1}^{N}\alpha_1(i)a_{ij}b_j(O_2)=c_2c_1\alpha_2(j)\end{aligned} \tag{10.44}$$

根据递推式(10.41)和式(10.42)可以证明下式成立：

$$\hat{\alpha}_t(j)=c_tc_{t-1}c_{t-2}\cdots c_1\alpha_t(j) \tag{10.45}$$

即

$$\alpha_t(j)=\hat{\alpha}_t(j)(c_1c_2\cdots c_t)^{-1} \tag{10.46}$$

由于前向概率 $P(O\mid\lambda)=\sum_{i=1}^{N}\alpha_T(i)$

用修正递推值表示为：

$$P(O\mid\lambda)=\sum_{i=1}^{N}(c_1c_2\cdots c_T)^{-1}\hat{\alpha}_T(i) \tag{10.47}$$

根据式(10.38)可得

$$\sum_{i=1}^{N}\hat{\alpha}_T(i)=c_T\sum_{i=1}^{N}\bar{\alpha}_T(i)=1 \tag{10.48}$$

所以有

$$P(O|\lambda)=(c_1c_2\cdots c_T)^{-1} \tag{10.49}$$

(2) 对后向变量进行定标

同上，我们引入两个变量，即递推值 $\bar{\beta}_t(j)$ 和修正递推值 $\hat{\beta}_t(j)$。

初始化

$$\bar{\beta}_T(j)=\beta_T(j)=1 \tag{10.50}$$

令

$$\hat{\beta}_T(j)=c_T\bar{\beta}_T(j) \tag{10.51}$$

同理类似于前向概率的定标最终可得

$$\beta_t(j)=(c_Tc_{T-1}\cdots c_t)^{-1}(\hat{\beta}_t(j)) \tag{10.52}$$

加入定标算法后(即用修正递推值代替原来的前后向变量)改写三个参数重估公式：

根据式(10.20)和式(10.46)可得

$$\bar{\pi}_i=\gamma_1(i)=\frac{\alpha_1(i)\beta_1(i)}{\sum_{i=1}^{N}\alpha_1(i)\beta_1(i)}=\frac{c_1^{-1}\hat{\alpha}_1(i)(c_Tc_{T-1}\cdots c_1)^{-1}\hat{\beta}_1(i)}{\sum_{i=1}^{N}c_1^{-1}\hat{\alpha}_1(i)(c_Tc_{T-1}\cdots c_1)^{-1}\hat{\beta}_1(i)} \tag{10.53}$$

又根据式(10.49)得到 π 定标后的重估公式

$$\overline{\pi}_i = \frac{\hat{\alpha}_1(i)\hat{\beta}_1(i)P(O \mid \lambda)c_1^{-1}}{\sum_{i=1}^{N}\hat{\alpha}_1(i)\hat{\beta}_1(i)P(O \mid \lambda)c_1^{-1}} \tag{10.54}$$

同理,式(10.36)变为

$$\begin{aligned}\overline{A}_{ij} &= \frac{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)(c_1c_2\cdots c_t)^{-1}a_{ij}b_j(O_{t+1})\beta_{t+1}(j)(c_{t+1}c_{t+2}\cdots c_T)^{-1}/P(O/\lambda)}{\sum_{t=1}^{T-1}\alpha_t(i)(c_1c_2\cdots c_t)^{-1}\beta_t(i)(c_tc_{t+1}\cdots c_T)^{-1}/P(O/\lambda)} \\ &= \frac{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)a_{ij}b_j(O_{t+1})\hat{\beta}_{t+1}(j)P(O \mid \lambda)/P(O \mid \lambda)}{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)\hat{\beta}_t(i)c_t^{-1}P(O \mid \lambda)/P(O \mid \lambda)}\end{aligned} \tag{10.55}$$

显然,上述式子中的概率值已被约去,所以最终的重估公式为

$$\overline{A}_{ij} = \frac{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)a_{ij}b_j(O_{t+1})\hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1}\hat{\alpha}_t(i)\hat{\beta}_t(i)c_t^{-1}} \tag{10.56}$$

式(10.37)定标后的推导如下。

同样,类似于前两个重估公式的推导,根据式(10.20)和式(10.46)

$$\overline{B}_j(O_k) = \frac{\sum_{\substack{t=1 \\ O_k=v_k}}^{T}\alpha_t(i)\beta_t(i)/P(O \mid \lambda)}{\sum_{t=1}^{T}\alpha_t(i)\beta_t(i)/P(O \mid \lambda)} = \frac{\sum_{\substack{t=1 \\ O_k=v_k}}^{T}\hat{\alpha}_t(i)\hat{\beta}_t(i)c_t^{-1}}{\sum_{t=1}^{T}\hat{\alpha}_t(i)\hat{\beta}_t(i)c_t^{-1}}$$

$$= \frac{\hat{\alpha}_1(i)\hat{\beta}_1(i)c_1^{-1} + [\hat{\alpha}_2(i)\hat{\beta}_2(i)c_2^{-1} \mid_{O_k=v_k} + \cdots + \hat{\alpha}_t(i)\hat{\beta}_t(i)c_t^{-1} \mid_{O_k=v_k} + \cdots + \hat{\alpha}_T(i)\hat{\beta}_T(i)c_T \mid_{O_k=v_k}]}{\hat{\alpha}_1(i)\hat{\beta}_1(i)c_1^{-1} + \hat{\alpha}_2(i)\hat{\beta}_2(i)c_2^{-1} + \cdots + \hat{\alpha}_t(i)\hat{\beta}_t(i)c_t^{-1} + \cdots + \hat{\alpha}_T(i)\hat{\beta}_T(i)c_T} \tag{10.57}$$

根据式(10.42)、式(10.43)有

$$\hat{\alpha}_t(j)c_t^{-1} = \sum_{i=1}^{N}\hat{\alpha}_{t-1}(i)a_{ij}b_j(O_t) \tag{10.58}$$

所以式(10.57)可以写为

$$\overline{B}_j(O_k) = \frac{\hat{\alpha}_1(i)\hat{\beta}_1(i)c_1^{-1} + \sum_{\substack{t=2 \\ O_k=v_k}}^{T}\sum_{i=1}^{N}\hat{\alpha}_{t-1}(i)a_{ij}b_j(O_t)\hat{\beta}_t(i)}{\hat{\alpha}_1(i)\hat{\beta}_1(i)c_1^{-1} + \sum_{t=2}^{T}\sum_{i=1}^{N}\hat{\alpha}_{t-1}(i)a_{ij}b_j(O_t)\hat{\beta}_t(i)} \tag{10.59}$$

前面给出了单个序列训练模型参数的重估公式。对于非特定人识别系统,如果语音的全部知识只是词汇表中每个单词的一个例词,却期望识别器具有非常优良的性能是不可能的,应该给识别器提供单词模式的各种变异情况。比较好的办法就是每个单词要有多个例词发音。

所以不能用一个观测序列来训练模型，为了有足够的数据来可靠地估计模型参数，必须使用多个观测序列。即每个模型参数都要使用多个样本来训练，假设有 L 个样本（对应于 L 个观测序列$[O^{(1)},O^{(2)},\cdots,O^{(L)}]$），现假定每个观测序列都是相互独立的，调整模型 λ 的参数以使 L 个 $P(O|\lambda)$ 乘积的值最大，此时对重估公式的修正办法是把每个观测序列的概率加在一起，这样修正后多序列的重估公式为

$$\overline{\pi}_i = \frac{\sum_{l=1}^{L}\hat{\alpha}_1^{(l)}(i)\hat{\beta}_1^{(l)}(i)P(O\mid\lambda)c_1^{(l)^{-1}}}{\sum_{i=1}^{N}\sum_{l=1}^{L}\hat{\alpha}_1^{(l)}(i)\hat{\beta}_1^{(l)}(i)P(O\mid\lambda)c_1^{(l)^{-1}}} \tag{10.60}$$

$$\overline{A}_{ij} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\hat{\alpha}_t^{(l)}(i)a_{ij}b_j(O_{t+1})\hat{\beta}_{t+1}^{(l)}(j)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\hat{\alpha}_t^{(l)}(i)\hat{\beta}_t^{(l)}(i)c_t^{(l)^{-1}}} \tag{10.61}$$

$$\overline{B}_j(O_k) = \frac{\sum_{l=1}^{L}\left[\hat{\alpha}_1^{(l)}(i)\hat{\beta}_1^{(l)}(i)c_1^{(l)^{-1}} + \sum_{t=2}^{T_l}\sum_{i=1}^{N}\hat{\alpha}_{t-1}^{(l)}(i)a_{ij}b_j(O_t)\hat{\beta}_t^{(l)}(i)\right]}{\sum_{l=1}^{L}\left[\hat{\alpha}_1^{(l)}(i)\hat{\beta}_1^{(l)}(i)c_1^{(l)^{-1}} + \sum_{t=2}^{T_l}\sum_{i=1}^{N}\hat{\alpha}_{t-1}^{(l)}(i)a_{ij}b_j(O_t)\hat{\beta}_t^{(l)}(i)\right]}_{O_k=v_k} \tag{10.62}$$

单序列和多序列 π_i 的重估公式中都出现了概率 P 的计算，这样又会引入新的下溢问题，解决办法是在迭代计算 P 的过程中，每次都乘以一个较大的数，这样分子分母每次都乘以一个相同的数，二者在同一数量级上，所以对重估公式没有影响。

10.2.3 隐马尔可夫模型用于语音识别

1. 实验方法

我们用 C++语言在 Windows 操作系统上实现了一个基于离散 HMM 的孤立词语音识别系统，实验的方框图如图 10.4 所示。对于一个孤立词识别系统，输入的语音信号是一个个的孤立单词。系统共使用了 50 词 16 个人的不同信噪比的语音数据来做实验（包括无噪声、15dB、20dB、25dB、30dB 的数据），每人每个词发音 3 次，其中 9 人的语音数据（某种 SNR）用于训练模型，另外 7 个人的语音数据（同一 SNR 下的）用于识别，得到这种 SNR 下语音的识别结果。每个词的 HMM 参数使用 27 个样本（9 人×3 次）来训练，测试样本文件的数目依实验所用的词汇量而不同。

具体实验步骤如下。

第 1 步：特征提取

特征提取一般要解决两个问题：一是从语音信号中提取（或测量）有代表性的合适的特征参数（即选取有用的信号表示）；另一个是进行适当的数据压缩。语音的特征参数是分帧提取的，每帧特征参数一般构成一个矢量，因此语音特征是一个矢量序列。系统中前端语音信号的采样率为 11.025kHz，帧长 10ms，110 个样点，帧移为 5ms。使用的特征提取方法是过零峰值幅度，即 ZCPA 特征。每个单词的 ZCPA 特征经时间和幅度归一化处理后得到统一的 64×16（1024）维的语音特征矢量序列。

第 2 步：矢量量化

特征提取后得到的语音特征矢量序列的数据率一般会很高，不便于其后的进一步处理，因此有必要采用一定的编码方法对数据进行压缩。由于系统中后端的识别方法采用离散HMM，且单词的矢量维数较高，所以提取的特征需要经过矢量量化处理。矢量量化是一种很有效的数据压缩技术，矢量量化的过程中首先需要生成码书，本文使用LBG法来聚类生成码书。其中初始码书的生成采用随机法，即从聚类前的矢量空间中随机选取 N 个(码书尺寸)M 维(码矢维数)矢量数据作为初始码字。这种方法易于实现且码书训练时间短，故常用于语音识别。语音信号中提取出来的特征经过数据压缩后便成为语音的模式，显然，语音模式是否具有代表性是语音识别成功与否的关键之一。首先将所有用来训练的单词特征组成一个大的语音特征矢量集(如50词矢量数目为 $50\times27\times(1024/4)$)，这个矢量集用来训练码书。系统采用的码书尺寸为128，码矢维数是4维，这些矢量被分到128个类中，每个聚类有一个标号(从1～128)。然后每个单词的1024维特征进入已训练好的矢量量化器，1024个数值每4维一个形成256个矢量，按照最近邻准则，对256个矢量进行矢量量化，每个矢量用它所在类的标号表示(从1～128)。最后，用各单词的特征矢量被量化后形成的码矢标号代替原来的1024维矢量成为每个单词的语音模式作为下一步处理的输入信号。

第3步：训练隐马尔可夫模型

上一步处理得到的单词特征标号即为离散HMM的输入序列 $\{O_1,O_2,\cdots,O_T\}$。对于离散HMM，每个单词用一个HMM模型参数表示(即 $\lambda=(A,B,\pi)$)，每个单词用27个样本序列来训练。系统采用自左向右无跨越形式的HMM模型，每个单词的模型为5个状态(如图10.8所示)，图10.9给出了用离散HMM训练好的的某个单词模型参数的状态转移矩阵示例。此外，由于码书尺寸为128，故观测符号数目 $N=128$。

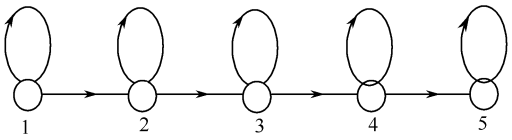


图10.8　无跨越由左向右模型

$$\boldsymbol{A}=\begin{bmatrix} 0.97922 & 0.02078 & 0 & 0 & 0 \\ 0 & 0.985461 & 0.014539 & 0 & 0 \\ 0 & 0 & 0.97241 & 0.02759 & 0 \\ 0 & 0 & 0 & 0.969138 & 0.030862 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

图10.9　状态转移矩阵

对于初值的设定，本系统模型中的三个参数均设定为等概率初值。训练方法使用经典的Baum-Welch算法，训练迭代终止条件为根据模型计算的一次发音概率值取对数(目的是避免下溢)后，在参数重估前后变化值的绝对值小于某个阈值(实验中选取0.01)。本系统每个单词模型参数均使用多个样本序列来训练，故笔者采用定标后的多序列重估式(10.60)，式(10.61)，式(10.62)来训练模型。实验结果证明本系统可以保持良好的收敛性，每个单词训练30次左右即可收敛。

第4步：对测试集单词进行识别

经过上一步的训练，每个单词都生成一套模型参数。类似于训练集数据的矢量量化，将用于测试的(即另外7个人在某种SNR的语音数据)每个单词的特征矢量送入第2步生成的矢量量化器。这样，每个单词也是一个码矢编号序列，每个序列都经过所有单词的HMM模型

参数计算概率值，这个过程采用 Viterbi 解码算法。这个算法使用的判据是寻找单个最佳状态序列（路径），即使 $P(Q|O,\lambda)$ 最大，这等效于使 $P(Q,O|,\lambda)$ 最大。这样概率最大值所对应的模型即为识别结果。最后用识别正确的单词数与所有测试集单词数做比值即为识别率。

2. 实验结果及讨论

表 10.1 为使用 ZCPA 特征和 HMM 的不同词汇量单词在各种 SNR 下的识别结果比较。

表 10.1　基于 ZCPA 特征用 HMM 所得识别率比较(%)

SNR(dB)	15	20	25	30	clean
10 词	85.7	84.7	86.2	85.7	89.1
20 词	76.6	81.2	82.4	81.7	85.7
30 词	77.1	81.9	83.1	82.9	83.5
40 词	76.6	79.0	81.3	82.6	83.0
50 词	72.1	74.5	80.1	79.0	81.7

下面是关于系统性能影响因素的讨论。

(1) 矢量量化影响

由于系统使用的是离散隐马尔可夫模型方法，所以需要事先对每个单词的特征参数进行矢量量化，这样不可避免地会引入量化误差，所以应使用好的方法生成码书，以减小由此引起的失真，从而使系统性能所受影响尽可能减小。

(2) 初值设定影响

文中介绍的 HMM 训练方法（Baum-Welch 算法）本质上是一种梯度下降方法，在训练过程中有可能到达局部最小值。因此，初值的选取比较重要，好的初值可以避免局部极小问题。我们可以加入一定的优化方法来选取初值（如可采取人工免疫算法在某个初值设定区间中选取一组最优参数作为初值，再用 Baum-Welch 算法进行训练）。在离散 HMM 中，参数 B 对系统的性能有很大影响，超过了参数 A 和 π。所以也可以单独对参数 B 初值的选取采用一定的优化方法。

(3) 训练数据量的影响

连续隐马尔可夫模型需要较少的训练数据，但对于离散 HMM，要求的训练数据量较大。为了训练出可靠的参数模型，必须加大训练集的数据。当在训练集中又加入了 5 个人的语音数据（共 16 人数据），测试集数据量不变，分别对 10 词到 50 词的数据进行了无噪声及信噪比为 15dB、20dB、25dB、30dB、clean 条件下的实验，结果如表 10.2 所示。实验结果表明增加训练集的样本数后，与 9 人训练相比系统识别率有大幅度提高。

表 10.2　增加训练样本后基于 ZCPA 特征和 HMM 的识别率比较(%)

SNR(dB)	15	20	25	30	clean
10 词	88.0	88.7	90.7	91.3	92.0
20 词	86.0	87.7	90.3	89.3	91.7
30 词	84.2	87.3	89.1	89.6	90.4
40 词	82.8	87.7	88.7	90.7	90.8
50 词	81.7	85.6	87.7	86.7	89.3

(4) 输出概率矩阵的平滑问题

训练集的有限性使得训练完以后的 $\boldsymbol{B}$ 矩阵中有一些零元素，这些不合理的零概率会给识别带来一定的影响，解决这个问题有三种方法：基数法，距离法和同现法。实验中采用的是最简单的基数法，它是将 $\boldsymbol{B}$ 矩阵中小于某个给定最小值的元素 e（e 依据生成矩阵确定）赋给一个值 ε（ε 取 $10^{-4} \sim 10^{-6}$），然后修改 $\boldsymbol{B}$ 矩阵的其他元素使它满足约束条件：即在第 j 个状态下 $\sum_{k=1}^{M} b_j(k) = 1$。具体方法如下：

设 $\boldsymbol{B}=\{b_j(k)\}$ 的第 j 行中有 R_j 个零值，则作如下参数调整

$$b_j(k)=\begin{cases}(1-R_j\varepsilon)b_j(k), & b_j(k)\leqslant e\\ \varepsilon, & b_j(k)>e\end{cases} \tag{10.63}$$

经过实验得出：将 $\boldsymbol{B}$ 矩阵进行平滑处理后，对训练集内的数据做识别测试时（称为特定人识别）识别率随 ε 值的增大而下降，未进行平滑前训练集内数据的识别率为 100%，平滑处理后识别率略有下降，这是由于 ε 的设置改变了原有训练参数而引起的。而对测试集数据进行识别测试时（称为非特定人识别），识别率随 ε 值的增加而上升。说明对于测试集，ε 越小，适应能力越差。所以这种输出概率矩阵平滑方法只适用于 HMM 的非特定人识别。在我们前述的识别系统中，选取 $\varepsilon=10^{-4}$，结果表明识别率较没有进行输出矩阵平滑前增加了 10%左右。

第11章　语音增强

11.1　概　述

语音信号是人类传播信息和感情交流的重要媒介，是听觉器官对声音传媒介质的机械振动的感知，也是人类最重要、最有效、最常用、最方便的通信方式。然而，在通信过程中语音会不可避免地受到来自周围环境、传输媒介引入的噪声，通信设备内部电噪声、乃至其他讲话者的干扰，这些干扰最终将使接收到的语音信号并非纯净的原始语音信号，而是受噪声污染的带噪语音信号。这里的“噪声”定义为所需语音信号以外的所有干扰信号。干扰信号可以是窄带的或宽带的、白噪声的或有色噪声的、声学的或电学的、加性的或乘性的，甚至可以是其他无关的语音。由噪声导致的语音质量的下降会使许多语音处理系统的性能急剧恶化。例如，由于语音生成模型是低速率语音编码的基础，当语音受到噪声干扰时，提取的模型参数将很不准确，重建的语音质量急剧恶化。再如，语音识别系统在实验室环境中可获得相当好的效果，但在噪声环境中，尤其是在强噪声环境中使用时，系统的识别率将受到严重影响。在这些情况下，采用语音增强技术进行预处理，将有效地改善系统性能。

语音增强有着广泛的应用，因此寻求一种有效的算法对带噪语音信号进行处理以达到较高抗噪效果的研究意义很大。在一般情况下干扰信号是随机信号，要完全排除噪声是不现实的，语音增强的目标对收听人而言主要是改善语音质量，提高语音可懂度，减少疲劳感；对语音处理系统（识别器、声码器、手机）而言是提高系统的识别率和抗干扰能力。

有关抗噪声技术的研究及实时环境下的语音信号处理系统的开发，在国内外作为语音信号处理的重要研究课题，已经做了大量的研究工作，取得了丰富的研究成果。目前国内外的研究成果大体分为三类解决方法：第一类方法是采用语音增强算法，提高语音识别系统前端预处理的抗噪声能力，提高输入信号的信噪比；第二类方法是寻找稳健的语音特征作为特征参数，实验证明，这类参数对宽带语音具有较好的抗噪性；第三类方法是基于模型参数自适应的噪声补偿算法。这类方法可以引入语音和噪声的统计知识，提出具有一定环境稳健性的处理算法，并且在应用中基本与语音模型的短时平稳的假设一致，所以成为目前研究的热点。但是，目前的补偿算法通常只考虑到噪声环境是平稳的，在低信噪比语音以及非平稳噪声环境中的效果并不理想。

解决噪声问题的根本方法是实现噪声和语音的自动分离，尽管人们很早就有这种愿望，但由于技术的难度，这方面的研究进展不大。近年来，随着声场景分析技术和盲分离技术的研究发展，利用在这些领域的研究成果进行语音和噪声分离的研究取得了一些进展。

语音增强与语音信号处理理论有关，而且涉及到人的听觉感知和语音学。噪声来源众多，随应用场合不同而特性各异，因此难以找到一种通用的语音增强算法可以适用于各种噪声环境，必须针对不同环境下的噪声采取不同的语音增强策略。因此，要进行语音增强首先要了解语音特性、人耳感知特性和噪声特性。

11.2 语音感知特性和噪声特性

11.2.1 语音特性

1. 语音信号具有短时平稳性

语音是时变的、非平稳、非遍历的随机过程。语音发声是一个时变过程，很多因素造成了发声系统的时变性，例如声道的面积随着时间和距离改变，气流速度随着声门处压力变化而变化等。但是声道形状有相对稳定性，在一段时间内(10～30ms)，人的声带和声道形状是相对稳定的，可认为其特征是不变的，因而语音的短时谱具有相对稳定性，在语音分析中可以把语音信号分为若干分析帧，每一帧的语音可以认为是准稳定的。语音增强可以利用这种短时平稳性。

2. 语音信号可以分为浊音和清音

语音可以分为周期性的浊音和非周期性的清音。浊音和清音经常在一个音节中同时出现。浊音部分和音质关系密切，在时域上呈现出明显的周期性，在频域上有共振峰结构，而且能量大部分集中在较低频段内，是语音中大幅度高能量的部分；清音部分则没有明显的时域和频域特征，类似于白噪声，能量较小，在强噪声中容易被掩盖，但在较高信噪比时能提供较多的信息。在语音增强中，可以利用浊音的周期性特征，采用梳状滤波器提取语音分量或者抑制非语音信号，而清音则难以与宽带噪声区分。

3. 语音信号可以利用统计分析特征描述

作为一个随机过程，语音信号可以利用许多统计分析特征进行分析。但由于语音信号的非平稳、非遍历性，因此长时间时域统计特性对语音增强算法的意义不大。语音的短时谱幅度统计特征是时变的，只有当分析帧长趋于无穷大时，才能近似具有高斯分布。在高斯模型的假设中，可以认为傅里叶展开系数是独立的高斯随机变量，均值为零，而方差是时变的。在有限帧长时这种高斯模型只是一种近似的描述，可以作为分析的前提在宽带噪声污染的带噪语音增强中应用。

11.2.2 人耳感知特性

语音感知对语音增强研究有重要作用，人耳对语音的感知主要是通过语音信号频谱分量幅度获取的，对各分量相位则不敏感，对频率高低的感受近似与该频率的对数值成正比。人耳具有掩蔽效应，即一个较弱声音由于另外一个较强声音的出现而导致该较弱声音能被感知阈值掩蔽的现象。人耳除了可以感受声音的强度、音调、音色和空间方位外，还可以在两人以上的讲话环境中分辨出所需要的声音，这种分辨能力是人体内部语音理解机制具有的一种感知能力。人类的这种分离语音的能力与人的双耳输入效应有关，称为“鸡尾酒会效应”。语音增强的最终效果度量是人耳的主观感觉，所以在语音增强中可以利用人耳感知特性来减少运算代价。

11.2.3 噪声特性

噪声来源取决于实际的应用环境，因而可以说噪声特性变化无穷。根据与输入语音信号的关系，噪声可分为加性噪声和非加性噪声两类。对某些非加性噪声而言，可以通过一定的变换转换成加性噪声。例如乘性噪声可以通过同态变换转换为加性噪声。某些与信号相关的量

化噪声也可以通过伪随机噪声扰动的方法转换成与信号独立的加性噪声。语音处理中的加性噪声大体上可以分为周期性噪声、脉冲噪声、宽带噪声和同声道其他语音的干扰等。

1. 周期性噪声

周期性噪声主要来源于发动机等周期性运转的机械，电气干扰也会引起周期性噪声。其特点是频谱上有许多离散的线谱。实际信号受多种因素的影响，线谱分量通常转变为窄带谱结构，而且通常这些窄带谱都是时变的，位置也不固定。必须采用自适应滤波的方法才能有效地区分这些噪声分量。

2. 脉冲噪声

脉冲噪声来源于爆炸、撞击、放电及突发性干扰等。其特征是时间上的宽度很窄。消除脉冲噪声通常可以在时域内进行，其过程如下：根据带噪语音信号幅度的平均值确定阈值。当信号超出这一阈值时判别为脉冲噪声。然后对信号进行适当的衰减，就可完全消除噪声分量，也可以使用内插方法将脉冲噪声在时域上进行平滑。

3. 宽带噪声

宽带噪声来源很多，热噪声、气流噪声及各种随机噪声源、量化噪声都可以视为宽带噪声。宽带噪声与语音信号在时域和频域上基本上重叠，只有在无语音期间，噪声分量才单独存在。因此消除这种噪声比较困难。对于平稳的宽带噪声，通常可以认为是白色高斯噪声。

4. 同声道语音干扰

干扰语音信号和待传语音信号同时在一个信道中传输所造成的语音干扰称为同声道语音干扰。区别有用语音和干扰语音的基本方法是利用它们的基音差别。考虑到一般情况下两种语音的基音不同，也不成整数倍，这样可以用梳状滤波器提取基音和各次谐波，再恢复出有用语音信号。

5. 传输噪声

这是传输系统的电路噪声。与背景噪声不同，它在时域是语音和噪声的卷积。处理这种噪声可以采用同态处理的方法，把非加性噪声变换为加性噪声来处理。

通过语音增强技术来改善语音质量的过程如图 11.1 所示。

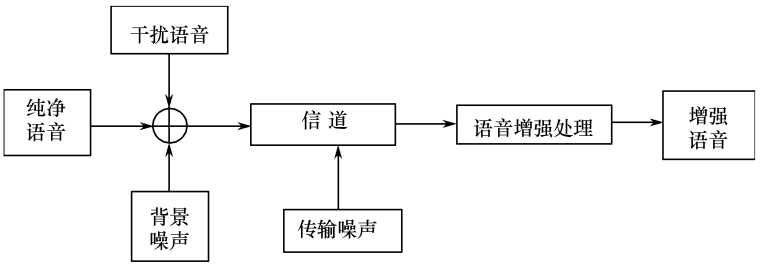


图 11.1　语音增强原理框图

11.3　语音增强算法

由于噪声特性各异，语音增强方法各有不同。多年来，人们针对加性宽带噪声研究了各种语音增强方法。目前应用的算法大致可以分为 4 种：参数方法、非参数方法、统计方法和其他方法。

11.3.1 参数方法

此类方法主要依赖于使用的语音生成模型(例如 AR 模型),需要提取模型参数(如基音周期、LPC 系数),常常使用迭代方法。如果实际噪声或语音条件与模型有较大的差距,或提取模型参数有困难,则此类方法容易失效。采用滤波器模型时,典型的有梳状滤波器、维纳滤波器、卡尔曼滤波器等。

在人类发声机理和语音产生的基本声学理论研究的基础上,建立起了离散时域的语音信号模型。语音信号被看成是线性时变滤波器在激励源激励下的输出。激励源分为浊音和清音两种,在浊音情况下,激励信号由一个周期脉冲发生器产生;在清音情况下,激励信号由一个随机噪声发生器产生;通常认为声道模型是一个全极点时变滤波器,滤波器参数可以通过线性预测分析得到。显然,如果能够知道激励参数和声道滤波器的参数,就能利用语音生成模型合成得到"纯净"的语音。这种方法的关键在于如何从带噪语音中准确地估计语音模型的参数(包括激励参数和声道参数)。

语音的全极点生成模型如图 11.2 所示,激励源为 $u(n)$,增益因子为 g,语音信号为 $s(n)$,全极点滤波器为 $H(z)=1/A(z)$,其中 $A(z)=1-\sum_{k=1}^{p}a_k z^{-k}$,$p$ 为阶数,a_k 为 LPC 系数。根据全极点模型有

$$s(n)=\sum_{k=1}^{p}a_k s(n-k)+gu(n) \tag{11.1}$$

其中,$s(n)$为清音时,$u(n)$为宽带噪声。$s(n)$为浊音时,$u(n)$为间隔是基音周期 T 的脉冲串。当不存在背景噪声时,干净的语音信号可以根据式(11.1)进行线性预测分析。存在背景噪声时,按照基本的自相关法或自协方差法已不能准确的求取线性预测系数。这时,基于 LPC 模型,可以使用最大后验概率估计法和卡尔曼滤波法等多种不同的估计方法。

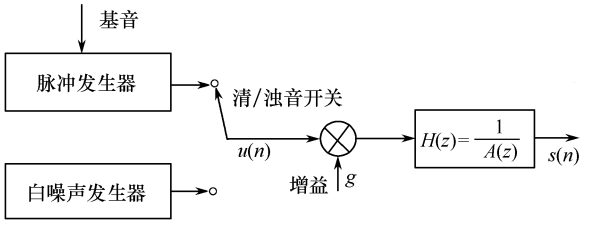


图 11.2 语音的全极点生成模型

利用语音信号浊音段有明显周期性的特点,可采用梳状滤波器提取语音分量来抑制噪声。滤波器输出信号是输入信号的延时加权和的平均值,当延时与信号的基音周期一致时,这个平均过程使周期性分量加强,而非周期分量或周期不同于信号的其他周期分量被抑制或消除。这种方法的关键是要准确估计出语音信号的基音周期。在基音变化的过渡段和强噪声背景干扰下无法精确估计基音周期时,这种方法的应用受到限制。

维纳滤波方法采用最小均方误差准则设计一个数字滤波器,带噪语音信号通过此滤波器便得到语音信号的估计。这个最佳滤波器就是维纳滤波器。维纳滤波器是在平稳条件的最小均方误差意义下的最优估计。但语音是非平稳的,实际环境中的噪声也是非平稳的,而且维纳滤波方法也没有完全利用语音生成模型。

卡尔曼滤波器是在已知状态方程和噪声统计特性的条件下,用线性预测分析参数实现波

形最小均方误差意义下的最佳估计器。卡尔曼滤波器弥补了维纳滤波器的两个缺陷，它是基于语音生成模型的，且在非平稳条件下也可以保证最小均方误差意义下的最优，故适合于非平稳噪声干扰下的语音增强。其缺点是：(1) 需要迭代估计模型参数，在噪声强时误差大；(2) 语音生成模型中假定激励是白噪声源，这仅对清音成立而对浊音是不成立的；(3) 计算量较大；(4) 优化标准是时域的波形误差最小，对语音信号而言此标准不够合理。

11.3.2 非参数方法

非参数方法不需要从带噪信号中估计模型参数，因此这种方法的应用范围较广。但由于没有利用可能的语言统计信息，故结果一般不是最优化的。这类方法包括自适应噪声抵消法、谱减法等。

1. 自适应噪声抵消法

图 11.3 为带自适应滤波器的噪声抵消器原理图，带噪语音输入为 $y(n)=s(n)+d(n)$，$s(n)$为语音信号，$d(n)$为未知噪声信号，$r(n)$参考噪声输入，也即自适应滤波器的输入，$v(n)$是该滤波器的输出。$r(n)$与 $s(n)$无关，而与 $d(n)$相关。自适应滤波器原理是，它在输入过程的统计特性未知或是输入过程的统计特性变化时，能够调整自己的参数，以满足某种最佳准则的要求。在图中，自适应滤波的目的就是通过对 $r(n)$的滤波，使输出的噪声估值 $v(n)$尽可能接近带噪语音中的 $d(n)$，然后从带噪语音中直接减去 $v(n)$，达到语音增强的目的。自适应滤波器通常采用 FIR 滤波器，系数采用最小均方误差(MMSE)准则来迭代估计。判断标准是使误差信号 $e(n)$能量最小：

$$e(n)=s(n)+d(n)-v(n)=s(n)+d(n)-\sum_{k=1}^{N}w_k r(n-k) \tag{11.2}$$

其中，w_k 是滤波器系数，N 是滤波器抽头数。MMSE 准则要求噪声和语音相互独立，这时，误差信号 $e(n)$能量最小，可保证 $v(n)$与 $d(n)$最接近。若噪声和语音不独立，则滤波器系数只能在无语音期间进行更新。

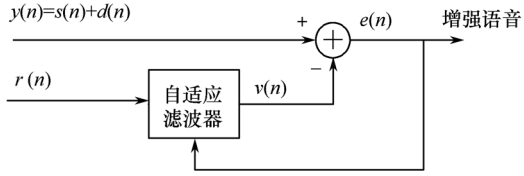


图 11.3 自适应噪声抵消原理图

目前，这种自适应噪声抵消法对含噪语音的增强效果最好，因为这种方法比其他方法多用了一个参考噪声作为辅助输入，从而获得了比较全面的关于噪声的信息，因而能得到更好的降噪效果。特别是在辅助输入噪声与语音中的噪声完全相关的情况下，自适应噪声抵消法能完全排除噪声的随机性，彻底抵消语音中的噪声成分，从而无论在信噪比方面还是在语音可懂度方面都能获得较大的提高。

2. 谱减法

谱减法是利用噪声的统计平稳性以及加性噪声与语音不相关的特点而提出的一种语音增强方法。这种方法没有使用参考噪声源，但它假设噪声是统计平稳的，即有语音期间噪声幅度谱的期望值与无语音间隙噪声的幅度谱的期望值相等。用无语音间隙测量计算得到的噪声频

谱的估计值取代有语音期间噪声的频谱，与含噪语音频谱相减，得到语音频谱的估计值。当上述差值得到负的幅度值时，将其置零。由于人耳对语音的感知主要是通过语音信号中各频谱分量幅度获得的，对各分量的相位不敏感。因此，此类语音增强方法将估计的对象放在短时谱幅度上。

假设带噪信号为

$$y(n)=s(n)+d(n),0\leqslant n\leqslant N-1 \tag{11.3}$$

其中，$s(n)$为纯净语音，$d(n)$为平稳加性噪声。$y(n)$通常需要加窗处理来消除分帧时带来的截断效应。这里为方便依然使用 $y(n)$表示加窗处理后的带噪信号。由于实际的分析帧长有限，傅里叶系数之间存在着一定的相关性。但为分析简便，我们仍假设傅里叶系数之间互不相关。设 $y(n)$的傅里叶变换为 $Y_k=|Y_k|\exp(\mathrm{j}\theta_k)$，$s(n)$的傅里叶变换为 $S_k=|S_k|\exp(\mathrm{j}\alpha_k)$，$d(n)$的傅里叶变换为 N_k，则有

$$Y_k=S_k+N_k \tag{11.4}$$

假设 $d(n)$满足高斯分布，其傅里叶变换 N_k 相当于多个高斯样本的加权和，仍然可以认为满足高斯分布，均值为 0，方差可以通过无语音期间的噪声分析得到。基于短时谱幅度估计的语音增强就是要利用已知的噪声功率谱信息，从 Y_k 中估计出 S_k。由于人耳对相位不敏感，因此只要估计出 S_k，然后利用带噪语音的相位，进行傅里叶反变换就可得到增强的语音。基于语音短时谱估计方法的一般原理如图 11.4 所示。根据实现估计的方法不同，可以分为谱减法、维纳滤波法和最小均方误差(MMSE)估计等，这里仅介绍谱减法。

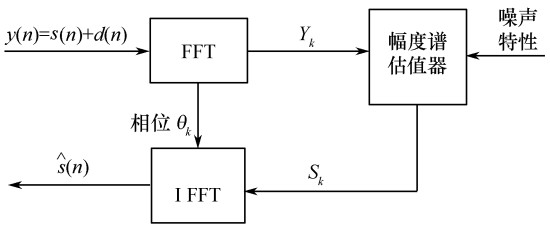


图 11.4　基于语音短时谱估计的原理框图

谱减法的基本原理图如图 11.5 所示。$y(n)$经 FFT 变换后，有 $Y_k=S_k+N_k$，由此可得

$$|Y_k|^2=|S_k|^2+|N_k|^2+S_kN_k^*+S_k^*N_k \tag{11.5}$$

由于 $s(n)$和 $d(n)$相互独立，所以 S_k 和 N_k 独立，而 N_k 为零均值的高斯分布，所以有

$$E||Y_k|^2|=E||S_k|^2|+E||N_k|^2| \tag{11.6}$$

对于一个分析帧内的短时平稳过程，有

$$|Y_k|^2=|S_k|^2+\lambda_n(k) \tag{11.7}$$

$\lambda_n(k)$为无语音时 $|N_k|^2$ 的统计平均值，由此可得原始语音的估计值

$$|\hat{S}_k|=[|Y_k|^2-E(|N_k|^2)]^{1/2}=[|Y_k|^2-\lambda_n(k)]^{1/2} \tag{11.8}$$

这里 $|\hat{S}_k|$是增强后的语音信号的幅度。

定义 $G_k=|\hat{S}_k|/|Y_k|$，及后验信噪比 $\gamma_k=|Y_k|^2/\lambda_n(k)$，式(11.8)可改写为

$$|\hat{S}_k|=G_k|Y_k| \tag{11.9}$$

$$G_k=(1-1/\gamma_k)^{1/2} \tag{11.10}$$

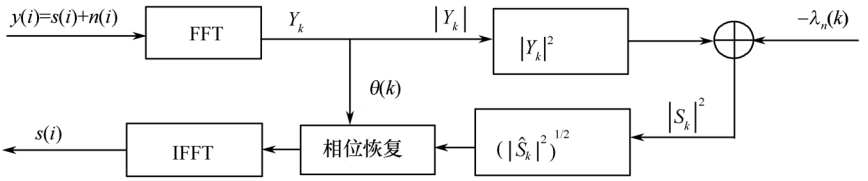


图 11.5　谱减法原理图

式(11.10)中当 γ_k 小于 1 时，将失去意义。因此，将式(11.10)改写为

$$G_k=\max(\varepsilon,(1-1/\gamma_k)^{1/2}) \tag{11.11}$$

其中，ε 是个大于零的常数。

从式(11.9)中可以清楚地看出谱相减的物理意义：它相当于对带噪语音的每一个频谱分量乘以一个系数 G_k。信噪比高时，含有语音的可能性大，衰减系数小。反之，则认为含有语音的可能性小，衰减系数大。

谱减法在频域将带噪语音的功率谱减去噪声的功率谱得到纯净语音功率谱估计，开方后就得到语音幅度谱估计，用带噪语音的相位来近似纯净语音的相位，再采用反傅里叶变换恢复时域信号。它的优点是比较简单，只需要进行正反傅里叶变换，而且实时实现较容易。但谱减法适用的信噪比范围较窄，在信噪比较低时对语音的可懂度损伤较大，这是因为信噪比主要代表了由浊音决定的大信号能量，而语音可懂度主要取决于元音和相对较小的代表辅音的信号。所以实际应用时除了要降低噪声外，还要兼顾语音的可懂度和自然度。另外，由于频谱直接相减会使增强后的语音产生“音乐噪声”，它具有一定的节奏性，听上去类似音乐声，由此而得名。

11.3.3　统计方法

统计方法较充分地利用了语音和噪声的统计特性，一般要建立模型库，需要训练过程获得初始统计参数，它与语音识别系统的联系很密切。如最小均方误差估计(MMSE)、利用听觉掩蔽效应等。

语音特性的分析告诉我们要了解语音短时谱幅度分布，可以通过两个途径：一是假设一个合理的概率分布模型；另一个则是通过实际统计的方法去获得。对于语音增强来说，听觉意义上的失真准则与给定噪声情况下语音频谱的后验分布是无法知道的，因此，对于特定的失真准则和后验概率不敏感的估计方法是很有用处的。

最小均方误差估计正是一种对特定的失真准则和后验概率不敏感的估计方法。它是利用已知的噪声功率谱信息，从带噪语音频谱分量中估计出纯净语音频谱分量，借助带噪语音相位得到增强的语音信号。考虑到大部分语音的变化是比较缓慢的，帧与帧之间的频谱有着一定的相似性，其相应频谱分量之间存在某种相关性，这种相关性可以反映在前一帧的频谱值对后一帧频谱的分布产生一种约束影响。由此，产生了基于帧间频谱分布约束的 MMSE 估计方法。人耳对声音强度的感受是与谱幅度的对数成正比的，同时，语音处理的实践也表明，采用对数失真准则更为适合一些。为此，将上述 MMSE 估计式进行推广，可得到频域分布约束下的短时对数谱的 MMSE 估计。

MMSE 算法达到了语音可懂度和清晰度的折中，适用信噪比的范围较广，但是由于需要统计各种参数，算法运算量大，实时性不好。

当两个能量不等的声音作用于人的听觉系统时，能量较高的信号可以使较低的信号不易

察觉，这就是人耳听觉系统的掩蔽效应。应用听觉掩蔽效应进行语音增强，语音信号能够掩蔽与其同时进入听觉系统的一部分能量较小的噪声信号，而使得这部分噪声不为人感知，利用一个功率谱域的基于听觉掩蔽门限的不等式准则，动态选择一个参数自适应变化的非线性函数估计语音短时谱幅度从而实现语音增强。这种方法在进行语音增强时，不需要把噪声完全抑制掉，只要使残留的噪声信号不被人感知即可，所以这样在消噪的同时可以减小不必要的语音失真。但是噪声掩蔽门限的计算是在纯净语音基础上得到的，在实际中一般要用带噪语音来估计掩蔽门限，这样估计的结果误差较大。

11.3.4 其他方法

其他方法包括小波变换、卡亨南－洛维变换(KLT)、离散余弦变换(DCT)、人工神经网络等。这些方法不像前三类方法那样成熟，可以概括地称为非主流方法。

利用之前的各种方法进行语音增强，需要知道噪声的一些特征或统计性质。在没有噪声先验知识的情况下，从唯一带噪语音信号中分离出语音信号，是非常困难的。小波变换能将信号在多个尺度上进行小波分解，各尺度上分解所得到的小波变换系数代表信号在不同分辨率上的信息。语音信号和噪声之间具有不同的 Lipschitz 指数，即信号具有正奇异性，而随机噪声具有负奇异性。这种性质在小波变换中，表现为信号的变换模值随尺度的增加而增加，随机噪声的变换模值随尺度的增加而减少。我们可以利用信号和噪声在小波下的这种截然不同的表现，提出一种有效的语音信号去噪方法。对输入带噪信号的小波系数设置一个合理阈值，仅让那些超过阈值的显著的小波系数用于小波逆变换来重构信号。这个阈值的选择确定对信号的去噪和恢复是有很重要影响的，因为这个门限阈值的确定直接影响信号去噪的效果和重构信号的失真程度。所以，用小波变换进行信号去噪，门限阈值的选择是关键。

KLT 用于语音增强，这种算法是把带噪语音沿着经过 KLT 变换的纯净语音向量空间进行分解，得到特征向量，修正每一个向量使得剩余噪声功率被限制在特定范围内，然后经 KLT 反变换合成输出增强后的语音。

离散余弦变换语音消噪方法与小波变换类似，通过对带噪信号进行离散余弦变换后用阈值函数处理，再进行离散余弦反变换就可以得到增强的语音信号。同样，阈值的选择是这类方法的关键，也是不断研究改进的重要内容。

语音增强方法可以看做从语音中区分出背景噪声的一种说话人区分方法。所以可以利用人工神经网络(例如 BP 网络)，用纯净语音信号作为网络训练信号，形成一个语音数据库，然后将带噪语音采样值与纯净语音采样值相比较并计算误差，基于误差最小准则利用 BP 算法调整网络权值，从而就可以提取出增强的语音信号。这种方法最适合语音识别领域。

上述各种方法各有优缺点，分别适用于不同情况。参数方法对语音的模型参数依赖性强，在低信噪比条件下不容易得到正确的模型参数；非参数方法由于频谱相减会产生“音乐噪声”；统计方法需要大量数据进行训练以得到统计信息；小波变换以及离散余弦变换的阈值选取困难，运算量大。

实际使用中常常根据具体的环境噪声和语音特性将不同方法结合起来应用，通过方法互补取得更好的语音增强效果。例如，MMSE 谱估计方法中需要用带噪语音估计出语音方差和噪声方差，这些未知参数可以用谱减法处理带噪语音部分得到的增强语音来计算。为了减少对语音可懂度的损伤，可以将 MMSE 谱估计法和听觉掩蔽效应结合。由于小波阈值语音增强算法中很难选取合适的阈值，一般达不到理想的效果，这时可以考虑将小波与 MMSE 相结合，

根据一定的均方误差准则自适应寻找到最佳阈值，或者基于神经网络选取合适阈值。还有，一般通过小波阈值消噪处理，达不到最佳语音增强效果，有必要在此基础上进一步对噪声进行处理，可以再通过 KLT 变换进行噪声处理，或者在小波处理之前先用谱减法进行预处理等。这样结合各种方法对语音进行增强就可以达到比较理想的效果。

11.3.5 谱减法语音增强的仿真实现

谱减法流程图如图 11.6 所示。

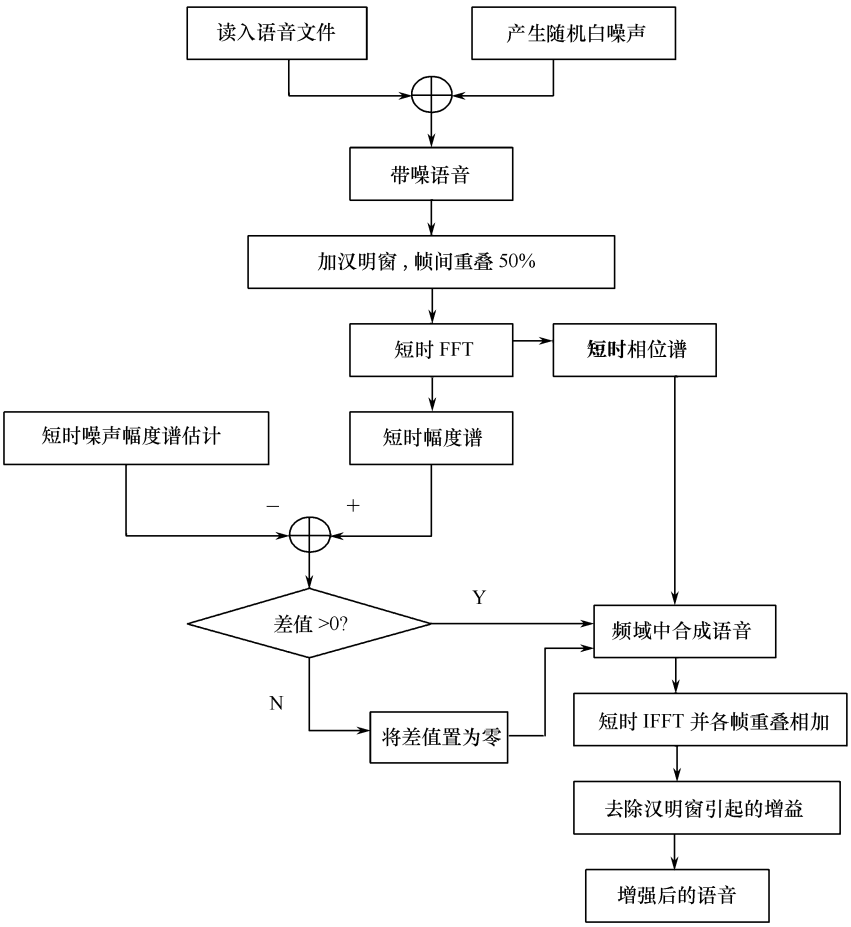


图 11.6 谱减法流程图

谱减法的 MATLAB 程序如下。

【程序 11.1】speechenhancement.m

```
clear all;
% ----------------------------读入语音文件----------------------------------------------------------
[speech,fs,nbits]=wavread('speech_clean1.wav');
                                                                    % 读入数据
% ----------------------------参数定义----------------------------------------------------------
winsize=256;                                                        % 窗长
n=0.04;                                                             % 噪声水平
```

```
size＝length(speech);                                                    % 语音长度
numofwin＝floor(size/winsize);                                           % 帧数
ham＝hamming(winsize)';                                                  % 产生汉明窗
hamwin＝zeros(1,size);                                                   % 定义汉明窗的长度
enhanced＝zeros(1,size);                                                 % 定义增强语音的长度
x＝speech'+ n* randn(1,size);                                            % 产生带噪信号
noisy＝n* randn(1,winsize);                                              % 噪声估计
N ＝ fft(noisy);                                                         % 对噪声傅里叶变换
nmag＝ abs(N);                                                           % 噪声功率谱
% ------------------------分帧-------------------------------------------------------------
for q＝1:2* numofwin- 1
frame＝x(1+ (q- 1)* winsize/2:winsize+ (q- 1)* winsize/2);
                                                                         % 对带噪语音帧间重叠一半取值
hamwin(1+ (q- 1)* winsize/2:winsize+ (q- 1)* winsize/2)＝...
hamwin(1+ (q- 1)* winsize/2:winsize+ (q- 1)* winsize/2)+ ham;
                                                                         % 加窗
y＝fft(frame.* ham);                                                     % 对带噪语音傅里叶变换
mag ＝ abs(y);                                                           % 带噪语音功率谱
phase ＝ angle(y);                                                       % 带噪语音相位
% --------------------------幅度谱减-----------------------------------------------------------
for i＝1:winsize
if mag(i)- nmag(i)> 0
clean(i)＝ mag(i)- nmag(i);
else
clean(i)＝0;
end
end
% ------------------------% 在频域中重新合成语音-----------------------------------------------
spectral＝ clean.* exp(j* phase);
% ---------------------反傅里叶变换并重叠相加-------------------------------------------------
enhanced(1+ (q- 1)* winsize/2:winsize+ (q- 1)* winsize/2)＝...
enhanced(1+ (q- 1)* winsize/2:winsize+ (q- 1)* winsize/2)+ real(ifft(spec-
tral));
end
% ------------------------除去汉明窗引起的增益-----------------------------------------------
for i＝1:size
if hamwin(i)＝＝0
enhanced(i)＝0;
else
enhanced(i)＝enhanced(i)/hamwin(i);
end
end
% --------------------计算增强前后信噪比-------------------------------------------------------
```

```
SNR1 = 10* log10(var(speech')/var(noisy));
                                                          % 加噪语音信噪比
SNR2 = 10* log10(var(speech')/var(enhanced- speech'));    % 增强语音信噪比
wavwrite(x,fs,nbits,'noisy.wav');                         % 输出带噪信号
wavwrite(enhanced,fs,nbits,'enhanced.wav');               % 输出增强语音
% ----------------------------画波形-------------------------------------------------------
figure(1);
subplot(3,1,1);plot(speech');
title('原始语音波形');
xlabel('样点数');ylabel('幅度');
axis([0 2.5* 10^4 - 0.3 0.3]);
subplot(3,1,2);plot(x);
title('加噪语音波形');
xlabel('样点数');ylabel('幅度');
axis([0 2.5* 10^4 - 0.3 0.3]);
subplot(3,1,3);plot(enhanced);
title('增强语音波形');
xlabel('样点数');ylabel('幅度');axis([0 2.5* 10^4 - 0.3 0.3]);
```

程序运行结果如图 11.7 所示。

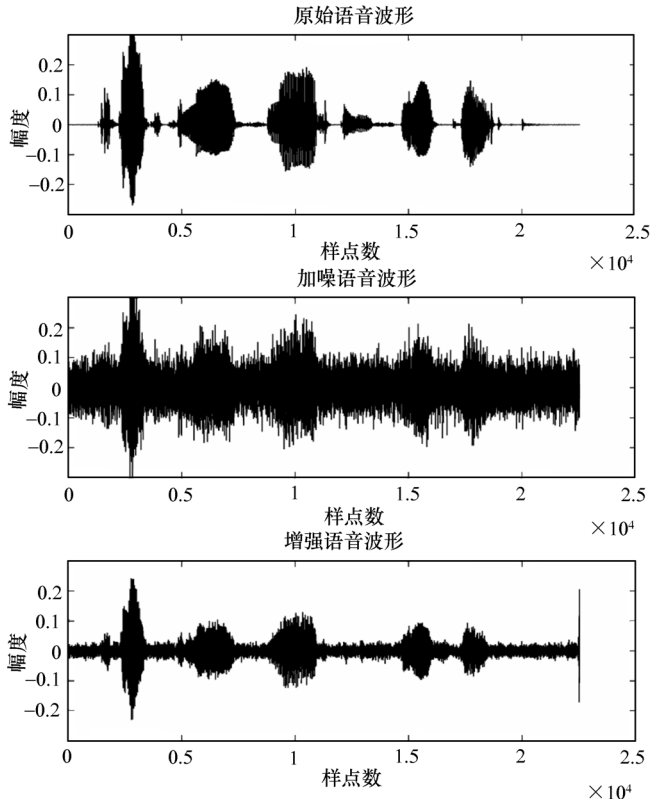


图 11.7　程序运行结果图

第 12 章　语音处理的实时实现

12.1　概　述

在实际的语音处理应用中，语音处理系统需要按照实时方式进行工作。近年来语音信号处理的技术水平不断提高，处理算法也日益复杂，语音处理实时系统对硬件提出了更高的要求。

语音信号处理的实时实现方式一般有以下几种：

① 在通用计算机（微型机、小型机或工作站）中插上专用的数字信号处理板，这种方式通常称为主从系统方式，适用于实验室环境下进行语音信号处理技术的研究。

② 用通用的单片机实现，这种方法可用于一些不太复杂的语音信号处理。

③ 由通用或专用 DSP 芯片以及其他辅助芯片构成一个独立的工作系统，也就是脱机式系统方式。

④ 用嵌入式系统构成独立的语音信号处理系统，可用于便携式语音处理实时系统，近年来嵌入式系统和 DSP 相结合的复合系统已经进入实用阶段。

20 世纪 80 年代初，随着世界上第一片可编程 DSP 芯片的诞生，DSP 应用系统得到了飞速发展。实际应用中，基于 DSP 的信号处理实时系统已得到广泛应用。几十年来，DSP 芯片已经在语音编码、语音合成、语音识别、语音增强、说话人辨识等语音实时处理领域得到了广泛的应用。近些年，嵌入式 DSP 实时系统由于其体积小、功耗低及便携性等优点，已逐渐成为研究热点。这些都极大地推动了语音处理技术的实用化发展。

本章简要介绍可编程 DSP 芯片的基本概念、基本结构和开发工具，对基于 DSP 芯片的实时语音处理系统的构成做了介绍，最后介绍了一个基于通用 DSP 芯片的实时语音处理系统的开发。

12.2　可编程 DSP 芯片应用基础

12.2.1　DSP 的发展历程

DSP 包含两个相关的概念，数字信号处理（Digital Signal Processing）和数字信号处理器（Digital Signal Processor），后者是在模拟信号变换成数字信号以后进行高速实时处理的专用信号处理器，在本章内容中，DSP 代表后者。

随着大规模集成电路技术的发展，1982 年世界上诞生了首枚 DSP 芯片（1978 年 AMI 公司发布的 S2811）。这种 DSP 器件采用微米工艺 NMOS 技术制作，虽功耗和尺寸稍大，但运算速度却比 MPU（微处理器）快了几十倍，在语音合成和编码解码器中得到了广泛应用。随着 CMOS 技术的进步与发展，第二代基于 CMOS 工艺的 DSP 芯片应运而生，其存储容量和运算速度成倍提高，成为语音处理、图像硬件处理技术的基础。20 世纪 80 年代后期，第三代 DSP

芯片问世，运算速度进一步提高，其应用范围逐步扩大到通信、计算机领域。20 世纪 90 年代 DSP 发展最快，相继出现了第四代和第五代 DSP 器件。第五代产品与第四代相比，系统集成度更高，将 DSP 内核及外围组件综合集成在单一芯片上。这种集成度极高的 DSP 芯片不仅在通信、计算机领域大显身手，而且逐渐渗透到与信号处理、自动控制等相关的应用领域。近些年，随着集成电路技术的进一步发展，嵌入式微处理器的应用也越来越广泛。TI 公司推出的 OMAP 系列及达芬奇系列芯片，其上集成了 ARM 和 DSP 双核 CPU，性能更完善，处理功能更强大，代表了新一代 DSP 的发展方向。

12.2.2 DSP 芯片的特点

DSP 芯片是一种特别适合于进行数字信号处理的微处理器，主要应用是实时快速地实现各种数字信号处理算法。根据数字信号处理的要求，DSP 芯片一般有以下一些主要特点：

① DSP 具有多总线结构，程序空间与数据存储空间分开，各有独立的地址总线和数据总线，取指令和读数据可以同时进行。

② DSP 具有独立的硬件乘法器，乘法指令可在单周期内完成，使卷积、数字滤波、FFT、相关运算、矩阵运算等算法中的大量乘法运算速度加快。

③ 采用流水作业，每条指令的执行划分为取指令、译码、取数、执行等若干步骤，由片内多个功能单元分别完成，相当于多条指令并行执行，大大提高了运算速度。

④ DSP 具有零消耗循环控制的专门硬件，零消耗循环是指处理器不用花时间测试循环计数器的值就能执行一组指令的循环，硬件完成循环跳转和循环计数器的衰减。

⑤ DSP 经常包含有专门的地址产生器，它能产生信号处理算法需要的特殊寻址，如循环寻址和位翻转寻址，循环寻址对应于流水 FIR 滤波算法，位翻转寻址对应于 FFT 算法。

⑥ DSP 片内具有快速 RAM，通常可通过独立的数据总线在两块中同时访问。

⑦ 快速的中断处理和硬件 I/O 支持。

12.2.3 DSP 芯片的分类

1. 按基础特性分

这是根据 DSP 芯片的工作时钟和指令类型来分类的。如果 DSP 芯片在某时钟频率范围内的任何频率上能正常工作，除计算速度有变化外，没有性能的下降，这类 DSP 芯片一般称之为静态 DSP 芯片。如果有两种或两种以上的 DSP 芯片，它们的指令集和相应的机器代码及引脚结构相互兼容，则这类 DSP 芯片称之为一致性的 DSP 芯片。

2. 按数据格式分

这是根据 DSP 芯片工作的数据格式来分类的。数据以定点格式工作的 DSP 芯片称为定点 DSP 芯片。以浮点格式工作的称为浮点 DSP 芯片。不同的浮点 DSP 芯片所采用的浮点格式不完全相同，有的 DSP 芯片采用自定义的浮点格式，有的 DSP 芯片则采用 IEEE 的标准浮点格式。

3. 按用途分

按照 DSP 芯片的用途来分，可分为通用型 DSP 芯片和专用型的 DSP 芯片。通用型 DSP 芯片适合普通的 DSP 应用，如 TI 公司的一系列 DSP 芯片。专用型 DSP 芯片是为特定的 DSP 运算而设计，更适合特殊的运算，如数字滤波，卷积和 FFT 等。

12.2.4 DSP 芯片的基本结构

在数字信号处理的运算中，常见的相关函数计算、卷积运算、信号滤波和各种变换算法大多可以归结为 $y=\sum a_i x_i$ 的乘加运算，因此 $y=x+a\times b$ 的形式出现最为频繁，所以 DSP 内部结构设计都是以优化上述乘加运算为主要目的。为了快速地实现数字信号处理运算，DSP 芯片一般都采用特殊的软硬件结构。DSP 芯片的基本硬件结构包括：哈佛结构、流水线操作、专用的硬件乘法器、特殊的 DSP 指令以及快速的指令周期。

1. 哈佛结构

哈佛结构是不同于传统的冯·诺依曼(Von Neuman)结构的并行体系结构，其主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个相互独立的存储器，每个存储器独立编址，独立访问。与两个存储器相对应的是系统中设置了程序总线和数据总线两条总线，从而使数据的吞吐率提高了一倍。而冯·诺依曼结构则是将指令、数据、地址存储在同一存储器中，统一编址，依靠指令计数器提供的地址来区分是指令、数据还是地址。取指令和取数据都访问同一存储器，数据吞吐率低。在哈佛结构中，由于程序和数据存储器在两个分开的空间中，因此取指和执行能完全重叠运行，进一步提高了运行速度和灵活性，如图 12.1所示。

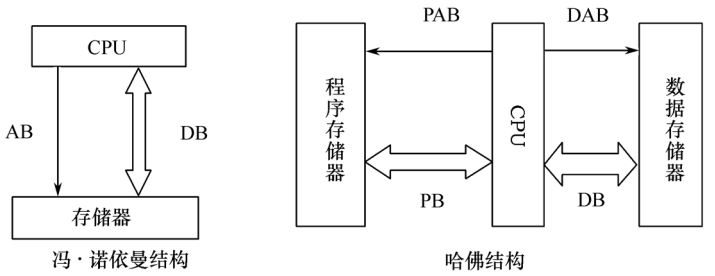


图 12.1　两种不同总线结构的比较

2. 流水线操作

流水线工作方式是指将指令分成几个不同的阶段，然后在不同时刻完成同一指令的不同阶段。DSP 大多采用流水技术，即将一条指令的执行过程分解成取指、译码、取数、执行等若干个阶段，每个阶段称为一级流水。每条指令都由片内多个功能单元分别完成取指、译码、取数、执行等操作，从而在不提高时钟频率的条件下减少了每条指令的执行时间。流水线可以分两种方式，一是 DSP 先对工作主频进行倍频，然后在每个倍频周期内完成指令的一个阶段，而整条指令在一个主频周期内完成，这种方式并不是真正意义上的流水线工作方式。二是保持主频不变，将一条指令的不同阶段分配在连续的几个指令周期内完成(TMS320 系列流水线深度从 2～6 级不等)，在一个指令周期内，执行不同指令的不同阶段，从而使每条指令真正独立运行的时间减少到最低。如图 12.2 所示，图中流水线深度为四级：取指、译码、取数、执行。

3. 专用的硬件乘法器

在一般形式的 FIR 滤波器中，乘法是 DSP 的重要组成部分。对每个滤波器抽头，必须做一次乘法和一次加法。乘法速度越快，DSP 处理器的性能就越高。在通用的微处理器中，乘法指令是由一系列加法来实现的，故需许多个指令周期来完成。相比而言，DSP 芯片的特征

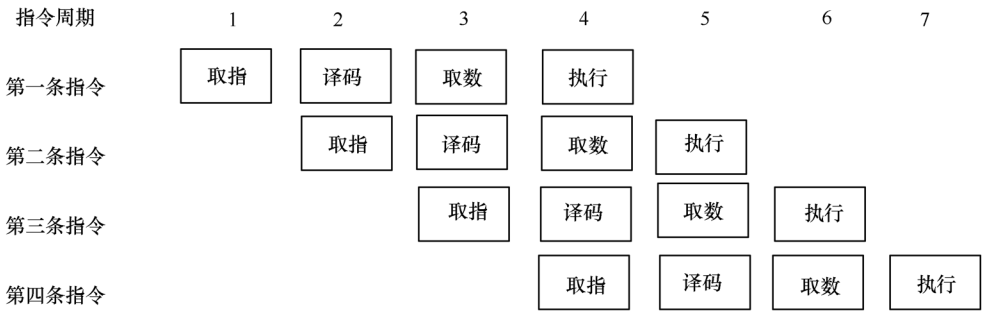


图 12.2　流水线示意图

就是有一个专用的硬件乘法器。在 TMS320 系列中，由于具有专用的硬件乘法器，乘法可在一个指令周期内完成。从最早的 TMS32010 实现 FIR 的每个抽头算法可以看出，滤波器每个抽头需要一条乘法指令 MPY 和三条其他指令：

```
LT      ;装乘数到 T 寄存器
DMOV    ;在存储器中移动数据以实现延迟
MPY     ;相乘
APAC    ;将乘法结果加到 ACC 中
```

其他三条指令用来将乘数装入到乘法器电路(LT)，移动数据(DMOV)以及将乘法结果(存在乘积寄存器 P 中)加到 ACC 中(APAC)。因此，若采用 256 抽头的 FIR 滤波器，这 4 条指令必须重复执行 256 次，且 256 次乘法必须在一个抽样间隔内完成。在典型的通用微处理器中，每个抽头需要 30～40 个指令周期，而 TMS32010 只需 4 条指令。如果采用特殊的 DSP 指令或采用 TMS320C54X 等新一代的 DSP 芯片，可进一步降低 FIR 抽头的计算时间。

4. 特殊的 DSP 指令

DSP 芯片的另一个特征是采用特殊的指令。DMOV 就是一个特殊的 DSP 指令，它完成数据移位功能。在数字信号处理中，延迟操作非常重要，这个延迟就是由 DMOV 来实现的。TMS32010 中的另一个特殊指令是 LTD，它在一个指令周期内完成 LT、DMOV 和 APAC 三条指令。LTD 和 MPY 指令可以将 FIR 滤波器抽头计算从 4 条指令降为 2 条指令。在第二代处理器中，如 TMS320C25，增加了 2 条更特殊的指令，即 RPT 和 MACD 指令，采用这 2 条特殊指令，可以进一步将每个抽头的运算指令数从 2 条降为 1 条：

```
RPTK    255         ;重复执行下条指令 256 次
MACD                ;LT、DMOV、MPY 及 APAC
```

5. 快速的指令周期

哈佛结构、流水线操作、专用的硬件乘法器、特殊的 DSP 指令再加上集成电路的优化设计，可使 DSP 芯片的指令周期在 200ns 以下。TMS320 系列处理器的指令周期已经从第一代的 200ns 降低至现在的 10ns 以下。快速的指令周期使得许多应用系统能够在 DSP 芯片上得到实时实现。

12.2.5　常用 DSP 芯片简介

在种类繁多的 DSP 芯片中，最成功的是美国得克萨斯仪器公司(TI)推出的一系列 DSP 产品。

1. TMS320C2000 系列

TMS320C2000 系列包括 TMS320C20x、TMS320C24x、TMS320C28x。

TMS320C20x 系列 DSP 芯片具有如下特点：

① 处理能力强。最高运算能力 40MIPS。

② 片内具有较大的闪存。不仅降低成本，减小体积，同时系统升级方便。

③ 功耗低。如使用 DSP 核的省电模式可进一步降低功耗。

④ 资源配置灵活。

TMS320C24x 系列针对数字控制系统应用做了优化设计。

2. TMS320C3x 系列

TMS320C3x 是 TI 的第三代产品，也是第一代浮点 DSP 芯片。该系列产品有 TMS320C30、TMS320C31、TMS320C32、TMS320C33 四种。

TMS320C31 是 TMS320C30 的简化和改进型，它在 TMS320C30 的基础上减去了一般用户不常用的一些资源，降低了成本，是一个性价比较高的浮点处理器，在国内已得到了较广泛的应用。TMS320C33 是 TMS320C3x 系列中性能最高，功耗最低的一种芯片，采用 3.3V 电压(核心电压 1.8V)，峰值功耗小于 200mW。集成 34KB 的双访问静态 RAM，具有一个串口，两个 32 位定时器，一个 DMA 控制器，两个外部标志输出 XF0 和 XF1。EDGEMODE 引脚可以选择外部中断是电平触发还是边沿触发，芯片具有自引导功能，具有两种省电模式。

3. TMS320C5000 系列

TMS320C5000 系列 DSP 包括 TMS320C54x 和 TMS320C55x 两大类。这两类芯片软件完全兼容，所不同的是 TMS320C55x 具有更低的功耗和更高的性能。

TMS320C54 系列是为实现低功耗、高性能而专门设计的定点 DSP 芯片，主要特点：

① 运算速度快。运算速度为 30～532MIPS。

② 优化的 CPU 结构。可高效地实现无线通信系统的各种功能。

③ 低功耗方式。可以节省 DSP 的功耗，特别适用于无线通信设备。

④ 智能外围设备。除了标准的串行口和时分复用(TDM)串行口外，多了 BSP、McBSP 即多通道缓冲串口(可与多达 128 个通道收发通信)和主机接口 HPI(可与外部标准的微处理器直接接口)。

TMS320C55x 是目前功耗最低的，与 TMS320C54x 兼容且性能比之提高了 5 倍，功耗仅为其 1/6。TMS320C55x 采用变指令长度提高代码效率，增强并行机制提高循环效率。TMS320C55x 以极低的功耗和优越的性能可以在通信、消费电子等很多领域得到广泛应用。

4. TMS320C6000 系列

TMS320C62x 系列是 1997 年开发的定点 DSP 芯片，特点是：

① 运行速度快。指令周期最小为 3.3ns，运算能力 2400MIPS。

② 内部集成了高度正交的两个乘法器和 6 个算术逻辑单元，单指令周期最大支持 8 条 32 位的指令。

③ 指令集不同。采用超长指令字结构，可一个时钟周期内并行执行多条指令。

④ 大容量的片内存储器和大范围的寻址能力。

⑤ 智能外围设备。

⑥ 低廉的使用成本。适用无线基站、无线 PDA、组合 Modem、GPS 等需要大运算能力的应用场合。

TMS320C64x 是 TMS320C6000 系列 DSP 中的最新的高性能定点芯片，其软件与

TMS320C62x 完全兼容。TMS320C64x 采用 VelociTI.2 结构的 DSP 核，增强的并行机制可以在单周期内完成 4 个 16×16 位或 8 个 8×8 位乘累加操作。采用两级缓冲(Cache)机制，第一级中程序和数据各用 16KB，而第二级中程序和数据共用 128KB。增强的 32 通道 DMA 控制器具有高效的数据传输引擎，可以提供超过 2GB/s 的持续带宽。与 TMS320C62x 相比，TMS320C64x 的总体性能提高了 10 倍。

TMS320C67x 是 TI 公司继定点 DSP 芯片 TMS320C62x 系列后开发的一种新型浮点 DSP 芯片，特点是：

① 运行速度快。指令周期 6ns，峰值运算能力 1336MIPS，单精度运算可达 1G FLOPS，双精度运算可达 250M FLOPS。

② 硬件支持 IEEE 格式的 32bit 单精度和 64bit 双精度浮点操作。

③ 集成了 32×32 比特的乘法器，其结果可为 32bit 或 64bit。

④ TMS320C67x 的指令集在 TMS320C62x 的指令集基础上增加了浮点执行能力，可以看做 TMS320C62x 指令集的超集。TMS320C62x 指令可在 TMS320C67x 上运行，而无须任何改变。

TMS320C67x 系列适用于对运算能力和存储量有高要求的应用场合。TMS320C64x 是 TMS320C6000 系列中最新的高性能定点芯片，软件与 TMS320C62x 完全兼容。

12.2.6 DSP 芯片的应用

DSP 芯片诞生之始，主要应用在数字信号处理领域。随着 DSP 技术的不断发展，其应用领域逐步扩展到自动化、机械电子等领域。目前，其应用范围已经越来越广。

① 作为通用数字信号处理器，可用于数字滤波、卷积、相关、FFT、希尔伯特变换、自适应滤波、窗函数产生、波形发生等。

② 在通信设备中，可用于高速调制解调器、编/译码器、自适应均衡器、传真、程控交换机、蜂窝移动电话、数字基站、数字留言机、回音消除、噪声抑制、电视会议、保密通信、卫星通信、TDMA/FDMA/CDMA 等各种通信制式。随着互联网络的迅猛发展，DSP 又在网络管理/服务、信息转发、IP 电话等新领域扮演着重要角色，而软件无线电的提出和发展进一步增强了 DSP 在无线通信领域的作用。

③ 在语音处理领域，可用于语音编码、语音合成、语音识别、语音增强、说话人辨识、说话人确认、语音储存等。

④ 在图形/图像处理领域，可用于三维图像变换、模式识别、图像增强、动画、电子出版、电子地图等。

⑤ 在自动控制领域，可用于磁盘、光盘、打印机伺服控制、发动机控制、电机驱动等。

⑥ 在仪器仪表中，可用于测量数据谱分析、自动监测及分析、暂态分析、勘探、模拟试验。

⑦ 在医学电子领域，可用于助听器、CT 扫描、超声波、心脑电图、核磁共振、医疗监护等。

⑧ 在军事与尖端科技领域，可用于雷达和声呐信号处理、雷达成像、自适应波束合成、阵列天线信号处理、导弹制导、火控系统、战场 C3I 系统、导航、全球定位 GPS、目标搜索跟踪、尖端武器试验、航空航天试验、宇宙飞船、侦察卫星。

⑨ 在计算机与工作站中，可用于阵列处理机、计算加速卡、图形加速卡、多媒体计算机。

⑩ 在消费电子领域，可用于数字电视、高清晰度电视、图像/声音压缩解压器、VCD/DVD/CD 播放机、电子玩具、游戏机、数字留言/应答机、汽车电子装置、音响合成、住宅电子安全系统、家用计算机控制装置。

12.3 基于 DSP 的语音处理系统

12.3.1 基于 DSP 的实时语音处理系统的构成

一个典型的基于 DSP 芯片的实时语音处理系统如图 12.3 所示。

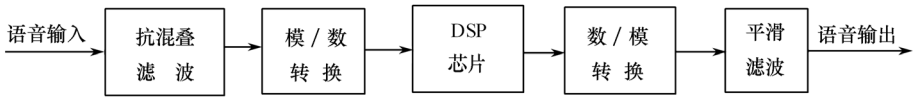


图 12.3 实时语音处理系统

输入语音信号首先进行带限滤波和采样，然后进行模数变换将语音信号变换成数字信号。DSP 芯片的输入是原始语音信号经过模/数转换后得到的数字化的语音信号，DSP 芯片对输入的数字信号进行某种形式的处理，如语音编码等，经过处理后的数字样值通过数/模转换(D/A)转换为模拟样值，最后再进行平滑滤波就可以得到连续的模拟波形。

图 12.3 中给出的基于 DSP 的语音处理系统模型只是一个典型模型，并不是所有的语音处理系统都必须具有模型中的所有部件。比如语音识别系统在输出端并不是连续的语音波形而是识别结果，如数字、文字等，因此可以不必进行数/模转换。

12.3.2 基于 DSP 的语音处理系统的特点

DSP 语音处理系统以数字信号处理为基础，因此具有数字处理的全部优点。

(1) 接口简单方便

由于数字信号的电气特性简单，不同的 DSP 系统相互连接时，在硬件接口上容易实现。在数据流接口上，各系统间只要遵循特定的标准协议即可。

(2) 编程方便，容易实现复杂的算法

在 DSP 系统中，DSP 芯片提供了一个高速计算平台，系统功能依赖于软件编程实现。当其与现代信号处理理论和计算数学相结合时，可以实现复杂的数字信号处理功能。DSP 语音处理系统中的可编程 DSP 芯片可使设计人员在开发过程中灵活方便地对软件进行修改和升级。

(3) 精度高，稳定性好

16 位数字系统可以达到 10^{-5} 的精度。数字信号处理仅受到量化误差和有限字长的影响，处理过程不引入其他噪声，因此具有较高的信噪比。另外，模拟系统的性能受到元器件参数性能影响比较大，而数字系统基本不变，因此数字系统更便于测试、调试及批量生产。

(4) 集成方便

现代 DSP 芯片都是将 DSP 内核及其外围电路综合集成在单一芯片上，这种结构便于设计便携式高集成度的数字产品。

12.3.3 基于 DSP 的语音处理系统的设计过程

图 12.4 是基于 DSP 的语音处理系统设计的一般过程。

在设计系统之前，必须根据语音处理系统要达到的目标和要求确定系统的各项性能指标。按照语音处理的要求，系统要达到的目标通常可用数据流程图、数学运算序列、正式的符号或

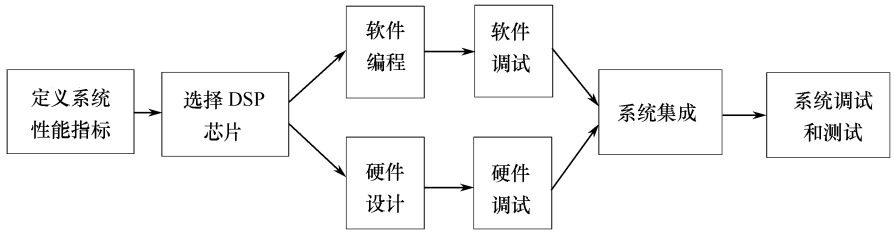


图 12.4　基于 DSP 的语音处理系统的设计流程

自然语言来加以详细描述。

然后要根据系统的要求进行算法模拟，为了实现系统的最终目标，需要对输入的语音信号进行适当的处理，而不同的系统性能要求采用不同的处理方法，要得到最佳的系统性能必须在这一步确定最佳的处理方法。在基于 DSP 的语音处理系统中，对语音信号的处理方法也就是语音信号处理算法。算法模拟输入的数据根据不同的情况可以是经过采样的实际信号，也可以是假设的数据。

接下来就要进行实时 DSP 语音处理系统的设计工作，实时 DSP 语音处理系统的设计包括硬件设计和软件设计两个并行的部分。硬件设计首先要根据运算量的大小、运算精度的要求、系统成本限制以及体积、功耗等要求选择合适的 DSP 芯片，然后根据系统要求和选好的 DSP 芯片设计外围电路及其他电路。软件设计和编程主要根据系统要求和所选的 DSP 芯片编写相应的 DSP 汇编程序或 C/C++程序。在实际应用系统中常采用高级语言和汇编混合编程的方法，即在算法运算量大的地方，用汇编语言，而运算量不大的地方则采用高级语言。采用这种方法，既可缩短软件开发的周期，提高程序的可读性和可移植性，又能满足系统实时运算的要求。

DSP 硬件和软件设计完成后，就需要进行硬件和软件的调试。软件的调试一般借助于 DSP 开发工具，如软件模拟器、DSP 开发系统或仿真器等。调试 DSP 算法时一般采用将实时结果与模拟结果进行比较的方法，如果实时程序和模拟程序的输入相同，则两者的输出应该一致。应用系统的其他软件可以根据实际情况进行调试。硬件调试一般采用硬件仿真器进行调试，如果没有相应的硬件仿真器，且硬件系统不是十分复杂，也可以借助于一般的工具进行调试。

系统的软件和硬件调试完成后，就可以将软件脱离开发系统而直接在应用系统上运行。当然，DSP 应用系统的开发，特别是软件开发是一个需要反复进行的过程，虽然通过算法模拟基本上可以知道实时系统的性能，但实际上模拟环境不可能做到与实时系统环境完全一致，而且将模拟算法移植到实时系统时必须考虑算法是否能够实时运行的问题。如果算法运算量太大不能在硬件上实时运行，则必须重新修改或简化算法。

12.4　DSP CCS 集成开发环境

12.4.1　DSP 的开发工具

可编程 DSP 芯片的开发需要一整套完整的软硬件开发工具。这些开发工具一般可以被分为代码生成工具和代码调试工具。代码生成工具的作用是将用 C 语言或汇编语言编写的

程序通过编译、汇编及链接，最后转化为可执行的 DSP 程序。TI 公司提供的代码生成工具主要包括：C 编译器、汇编器、链接器、文件格式转换程序、库存生成程序、文档管理程序、库存文件头文件等。代码调试工具的作用是对 DSP 程序进行调试以达到预定的设计目标。代码调试工具主要包括 C/汇编语言源码调试器、初学者工具 DSK、软件模拟器、评价模块 EVM、软件开发系统 SWDS、软件仿真器等。

CCS 是 TI 推出的集代码生成工具和代码调试工具于一体的 DSP 集成开发环境，结合仿真器等硬件调试工具，就可以进行几乎所有的 DSP 软硬件测试。

12.4.2 CCS 概述

为了缩短 DSP 的软件开发周期，1999 年 TI 公司推出了集成开发环境 CCS IDE（Code Composer Studio Integrated Development Environment）。它支持 TMS320C2000、C5000 和 C6000 系列。

CCS 是一个开放和具有强大集成能力的集成开发环境，该套开发环境集成代码生成工具和代码调试工具为一体，能完成 DSP 系统开发过程的各个环节，它由先进的开发工具组成直观的系统，可大幅度减小 DSP 编程时间，同时，它包括了高级的编码工具以及可供第三方接入的开放式结构。这种开放式结构使得开发人员可以采用特定的工具自定义环境满足特殊的设计需要。

CCS 提供了非常良好的用户界面，具有菜单、对话框式接口；具备丰富的图形图标，辅之以完整的可即时访问的在线帮助，使开发人员不必记忆复杂命令，就能够轻松地掌握和使用 CCS 开发环境。

1. 工程管理功能

CCS 对一个 DSP 应用系统的文件管理是通过工程方式进行的。对于熟悉 VC 程序开发的人员来说，工程方式管理是不难理解的。工程中包括着系统所有的源代码文件、目标代码文件、链接命令文件、配置文件、库函数、头文件等。采用工程的方式统一管理各种文件、非常直观、灵活，这是 CCS 不同于传统开发工具的一大改进。

① 向工程中添加文件，CCS 会根据文件的类型将其自动分配到相应的文件夹。

② 不用添加头文件，CCS 会自动搜索源文件用到的头文件，并添加到工程中。

2. 源代码编辑功能

① 对 C 语言和汇编语言的源代码进行编辑，同时设计者可以采用 C 语言和汇编语言混合显示模式，即在每一条 C 指令后显示相应的汇编语言指令。

② 能够在一个或多个文件中查找、替换、快速搜寻特定字符串。同时 CCS 编辑器下的一些常用命令，如文件的生成、打开、存储以及文本的剪切、粘贴等和常用编辑软件一样，便于掌握。

③ 用户可以根据自己的习惯定制不同的快捷方式。

④ 对关键字、注释、字符串等以不同的颜色高亮显示。

⑤ 高亮选定某一指令后，按下 F1 键，可以得到此指令的帮助。

3. 代码生成功能

在 CCS 开发环境下，每一个 DSP 系统所用到的源代码文件、目标文件、库函数、链接命令文件等都包含在相应的工程中。CCS 对某一应用系统的生成，实际上就是实现对这一工程的

编译、汇编和链接。

① 通过对话方式设置 Build 命令选项。

② C 编译器将 C 语言源代码编译成为汇编语言代码。

③ 扫描文件，为整个工程创建依赖关系树。

④ 运行支持库，包括 C 编译器所支持的 ANSI 标准运行支持函数、编译器公用程序函数和 C 编译器支持的 I/O 函数。

4. 代码调试功能

① 提供完善的程序运行控制的功能，如单步执行、条件执行和端点设置等。

② 综合数据显示能力，可以方便地通过不同窗口显示和修改变量、存储器和寄存器的值。

③ 在断点处自动更新所有窗口。

④ 显示反汇编文件和 C 文件，实现 C 语言和汇编语言源代码的同时调试。

⑤ 图形显示 DSP 中的数据。

⑥ 统计代码的执行时间。

除此之外，CCS 还提供 DSP/BIOS 插件，它不但支持可用于可视化的探测、跟踪和监视一个 DSP 应用程序的实时分析，甚至可以作为应用程序的 I/O 部分，用于从主机获取原始数据和向主机发送处理数据结果，因此可以大大节省整个软件系统的开发周期。

用 C 或汇编语言开发的应用程序，经代码生成工具编译、链接后生成可执行文件（*.out），还需要把可执行文件加载到指定的 DSP 目标板或软件模拟器中进行逻辑错误和实时性调试。CCS 支持的调试器包括：Simulator（软件模拟器）、EVM、DSK 板和硬件仿真器 XDS510 等。图 12.5 给出了 CCS 的功能框图。

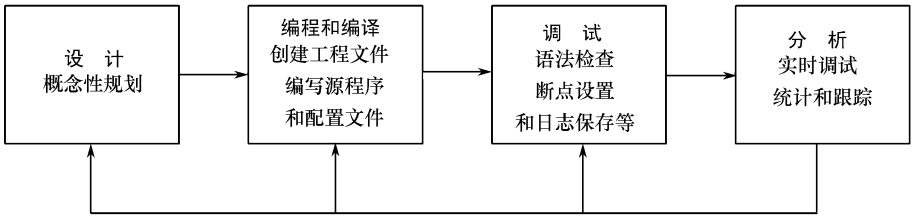


图 12.5 CCS 的功能框图

12.4.3 CCS 的构成

CCS 包括如下各部分：CCS 代码生成工具、CCS 集成开发环境（IDE）、DSP/BIOS 插件程序和 API 函数、RTDX 插件、主机接口和 API 函数。

CCS 构成及接口如图 12.6 所示。

1. CCS 的代码生成工具

代码生成工具奠定了 CCS 所提供的开发环境的基础。图 12.7 是一个典型的软件开发流程图，图中阴影部分表示通常的 C 语言开发途径，其他部分是为了强化开发过程而设置的附加功能。

图 12.7 描述的工具如下：

① C 编译器（C compiler）：产生汇编语言源代码，其细节参见 TMS320C54x 最优化 C 编译器用户指南。

② 汇编器（assembler）：把汇编语言源文件翻译成机器语言目标文件，机器语言格式为公

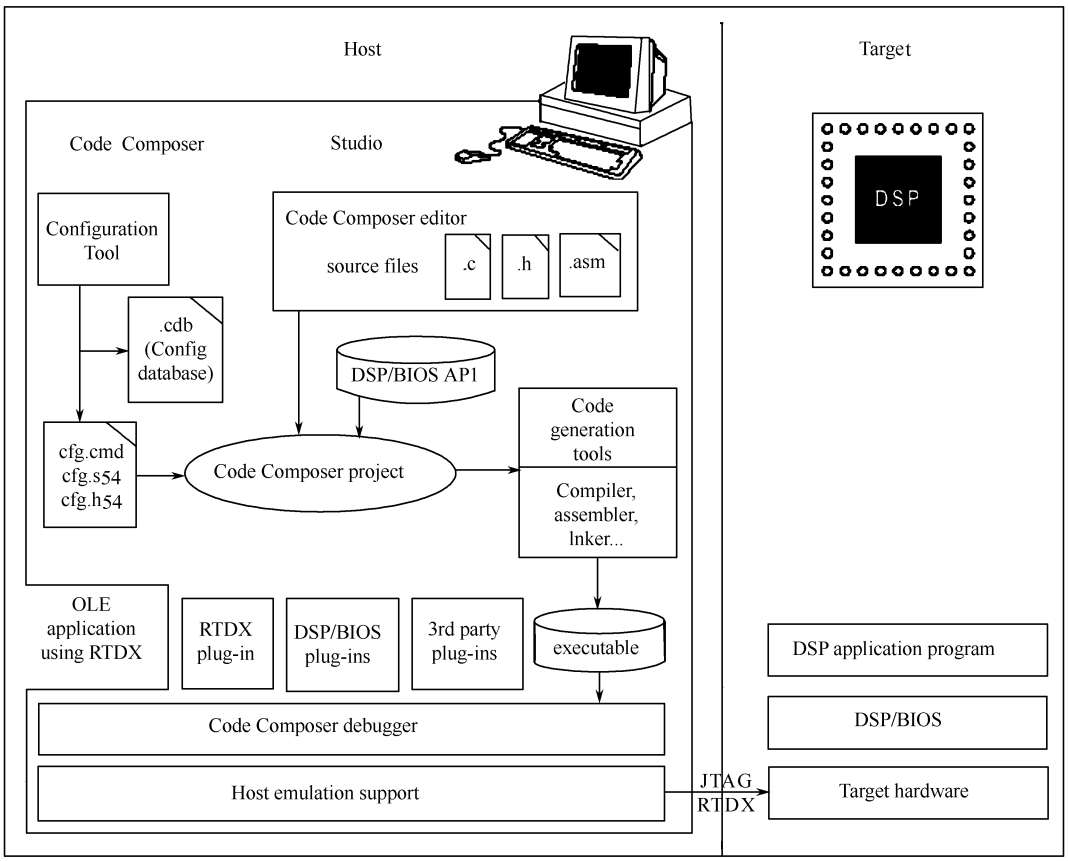


图 12.6　CCS 构成及接口

用目标格式(COFF),其细节参见 TMS320C54x 汇编语言工具用户指南。

③ 连接器(linker):把多个目标文件组合成单个可执行目标模块。它一边创建可执行模块,一边完成重定位以及决定外部参考。连接器的输入是可重定位的目标文件和目标库文件,有关连接器的细节参见 TMS320C54x 最优化 C 编译器用户指南和汇编语言工具用户指南。

④ 归档器(archiver):允许用户把一组文件收集到一个归档文件中。归档器也允许用户通过删除、替换、提取或添加文件来调整库,其细节参见 TMS320C54x 汇编语言工具用户指南。

⑤ 助记符到代数汇编语言转换公用程序(mnemonic to algebraic assembly translator utility):把含有助记符指令的汇编语言源文件转换成含有代数指令的汇编语言源文件,其细节参见 TMS320C54x 汇编语言工具用户指南。

⑥ 建库程序(library build utility):可以利用它建立满足自己要求的“运行支持库”,其细节参见 TMS320C54x 最优化 C 编译器用户指南。

⑦ 运行支持库(run-time-support libraries):它包括 C 编译器所支持的 ANSI 标准运行支持函数、编译器公用程序函数、浮点运算函数和 C 编译器支持的 I/O 函数,其细节参见 TMS320C54x 最优化 C 编译器用户指南。

⑧ 十六进制转换公用程序(hex conversion utility):它把 COFF 目标文件转换成 TI-Tagged、ASCII-hex、Intel、Motorola-S、或 Tektronix 等目标格式,可以把转换好的文件下载到

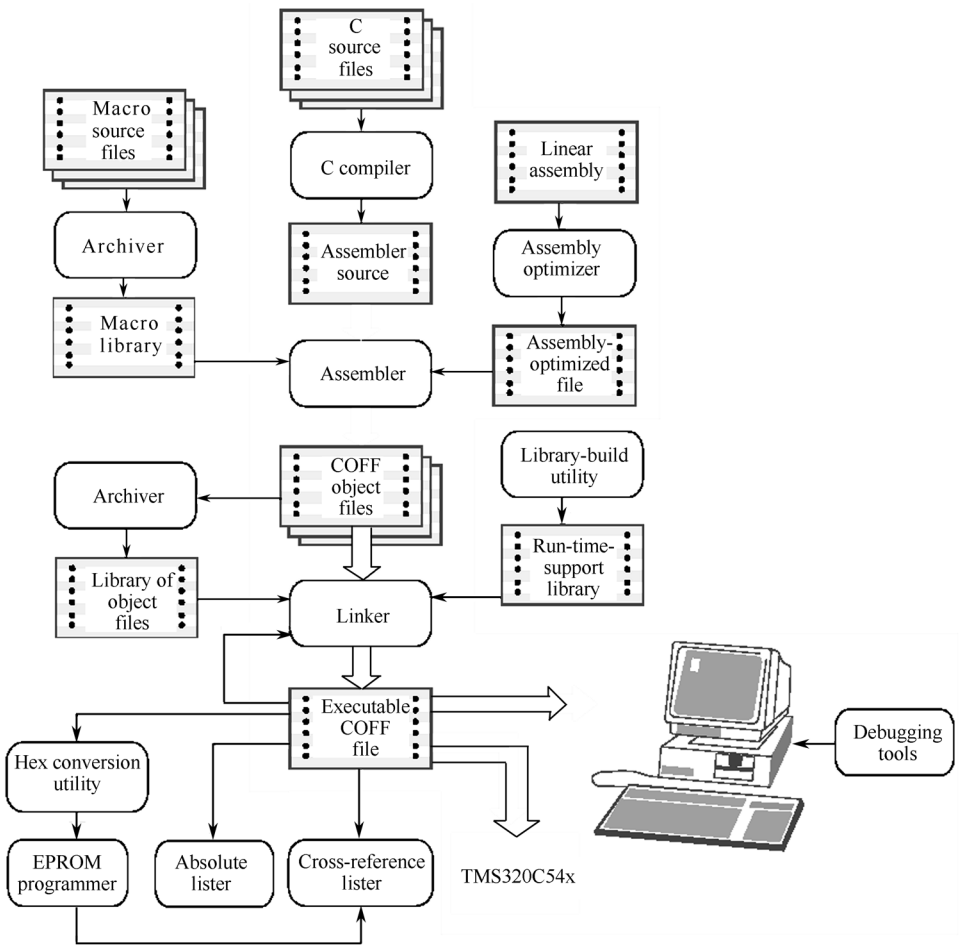


图 12.7　软件开发流程

EPROM 编程器中，其细节参见 TMS320C54x 汇编语言工具用户指南。

⑨ 交叉引用列表器(cross-reference lister)它用目标文件产生参照列表文件，可显示符号及其定义，以及符号所在的源文件，其细节参见 TMS320C54x 汇编语言工具用户指南。

⑩ 绝对列表器(absolute lister)：它输入目标文件，输出 .abs 文件，通过汇编 .abs 文件可产生含有绝对地址的列表文件。如果没有绝对列表器，这些操作将需要冗长乏味的手工操作才能完成。

2. CCS 集成开发环境

CCS 集成开发环境(IDE)允许编辑、编译、汇编、链接和调试 DSP 目标程序。

(1) 编辑源程序

CCS 允许编辑 C 源程序和汇编语言源程序，还可以在 C 语句后面显示汇编指令的方式来查看 C 源程序。集成编辑环境支持下述功能：用彩色加亮关键字、注释和字符串；以圆括号或大括号标记 C 程序块，查找匹配块或下一个圆括号或大括号；在一个或多个文件中查找和替代字符串，能够实现快速搜索；取消和重复多个动作；获得“上下文相关”的帮助；用户定制的键盘命令分配。

(2) 创建应用程序

应用程序通过工程文件来创建。工程文件中包括 C 源程序、汇编源程序、目标文件、库文件、连接命令文件和包含文件。编译、汇编和连接文件时,可以分别指定它们的选项。在 CCS 中,可以选择完全编译或增量编译,可以编译单个文件,也可以扫描出工程文件的全部包含文件从属树,也可以利用传统的 makefiles 文件编译。

(3) 调试应用程序

CCS 提供下列调试功能:设置可选择步数的断点;在断点处自动更新窗口查看变量;观察和编辑存储器和寄存器;观察调用堆栈;对流向目标系统或从目标系统流出的数据采用探针工具观察;并收集存储器映像;绘制选定对象的信号曲线;估算执行统计数据;观察反汇编指令和 C 指令;CCS 提供 GEL 语言,它允许开发者向 CCS 菜单中添加功能。

3. DSP/BIOS 插件

在软件开发周期的分析阶段,调试依赖于时间的例程时,传统调试方法效率低下。DSP/BIOS 插件支持实时分析,它们可用于探测、跟踪和监视具有实时性要求的应用例程。DSP/BIOS API 具有下列实时分析功能:程序跟踪(Program tracing)显示写入目标系统日志(target log)的事件,反映程序执行过程中的动态控制流;性能监视(Performance monitoring)跟踪反映目标系统资源利用情况的统计表,诸如处理器负荷和线程时序;文件流(File streaming)把常驻目标系统的 I/O 对象捆绑成主机文档。

DSP/BIOS 也提供基于优先权的调度函数,它支持函数和多优先权线程的周期性执行。

4. 硬件仿真和实时数据交换

TI DSP 提供在片仿真支持,它使得 CCS 能够控制程序的执行,实时监视程序运行。增强型 JTAG 连接提供了对在片仿真的支持,它是一种可与任意 DSP 系统相连的低侵扰式的连接。仿真接口提供主机一侧的 JTAG 连接,如 TI XDS510。为方便起见,评估板提供在板 JTAG 仿真接口。

在片仿真硬件提供多种功能:

- DSP 的启动、停止或复位功能。
- 向 DSP 下载代码或数据。
- 检查 DSP 的寄存器或存储器。
- 硬件指令或依赖于数据的断点。
- 包括周期的精确计算在内的多种记数能力。
- 主机和 DSP 之间的实时数据交换(RTDX)。

CCS 提供在片能力的嵌入式支持;另外,RTDX 通过主机和 DSP API 提供主机和 DSP 之间的双向实时数据交换,它能够使开发者实时连续地观察到 DSP 应用的实际工作方式。在目标系统应用程序运行时,RTDX 也允许开发者在主机和 DSP 设备之间传送数据,而且这些数据可以在使用自动 OLE 的客户机上实时显示和分析,从而缩短研发时间。

12.5 基于 TMS320C5409 的实时语音识别系统

12.5.1 硬件介绍

TMS320C54x 系列是 TI 为实现低功耗,高性能而专门设计的定点 DSP 芯片。本节以

TMS320C5409DSP 为核心并配置一些外围器件构成一个非特定人小词汇量语音识别系统。系统硬件主要由处理器模块、编解码模块、存储器模块、串行通信模块和电源模块等组成。TMS320C5409 是整个系统的核心，其他设备都围绕着它工作。系统硬件结构如图 12.8 所示。

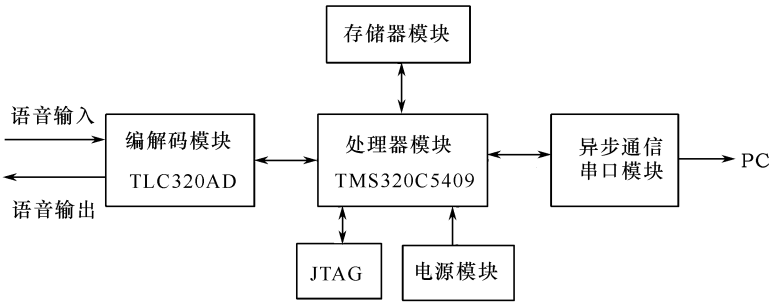


图 12.8　系统硬件结构

1. 处理器模块

(1) TMS320C5409 DSP 简介

TMS320C5409 数字信号处理器(简称 C5409)是 TI 公司生产的新一代定点 DSP 芯片，性价比极高，是目前定点 DSP 的主流产品。它具有一个 40bit 的算术逻辑运算单元，内含两个 40bit 的累加器和一个 40bit 的桶形移位器，能在单周期内完成 32bit 操作数的加/减法运算。C5409 片内有 8 组 16bit 总线(4 组数据或程序总线，4 组地址总线)，构成增强型哈佛结构的总线系统。指令按流水线方式执行，能在单周期内完成读两个操作数和写一个操作数的操作。为充分利用这种多总线结构和流水线操作的优点。TI 公司还专门开发了一套并行指令，能在单周期内执行一次存储/加载操作和一次算术运算。

C5409 内部有 16 千字(1 千字$=2^{10}\times16$bit$=1024\times16$bit)的 ROM 和 32 千字的 RAM，可以作为程序存储器或数据存储器，另外还有三个多通道缓冲串行口、一个 8 位并行增强型主机接口(HPI)、一个 16 位定时器、一个六通道 DMA 控制器和一个 PLL 时钟发生器。

(2) 处理器模块设计

处理器模块以 TMS320C5409 为核心芯片，主要完成对采样后的离散语音信号的识别处理及对其他模块的控制。

TMS320C5409 时钟的提供一般有两种途径：一种是使用 TMS320C5409 内部所提供的晶振电路，将一个晶体跨接在 TMS320C5409 芯片的引脚 X1 和引脚 X2/CLKIN 之间，使内部振荡器工作；另一种方法是使用外部时钟，将一个外部时钟源直接连接到 X2/CLKIN 引脚，X1 引脚悬空。本系统采用的是第一种方法，即使用内部振荡电路来提供时钟。

TMS320C5409 内部具有一个可编程锁相环(PLL)，它具有频率放大和时钟信号的提纯作用。具有软件可编程 PLL 的 DSP 器件可以选用两种时钟方式来配置：一种为 PLL 模式(即输入时钟乘以从 0.25～15 共 31 个系数之一)，另一种为 DIV(分频器)模式(即输入时钟除以 2 或 4)。本系统采用第一种方式来配置。

软件可编程 PLL 受时钟模式寄存器 CLKMD 控制，通过设定三个输入引脚 CLKMD1、CLKMD2 和 CLKMD3 来确定。TMS320C5409 时钟方式的配置方法如表 12.1 所示。

2. 编解码模块

编解码模块以 TLC320AD50C 为核心，主要完成对从麦克风或线性音频接口输入的语音

信号的采样和模数转换，并能够将 DSP 处理后的信号转换成模拟信号由音频输出口输出。

表 12.1　时钟方式的配置方法

CLKMD1	CLKMD2	CLKMD3	CLKMD 复位值	时钟工作模式
0	0	0	E007h	PLL×15
0	0	1	9007h	PLL×10
0	1	0	4007h	PLL×5
1	0	0	1007h	PLL×2
1	1	0	F007h	PLL×1
1	1	1	0000h	1/2(PLL 无效)
1	0	1	F000h	1/4 (PLL 无效)
0	1	1	…	保留

TLC320AD50C(简称 AD50)是 TI 公司生产的多媒体音频编解码器芯片，它为系统提供了一个灵活、通用的音频前端。该芯片集成了 A/D 和 D/A，最高采样频率为 22.05kHz。A/D 和 D/A 的转换精度均 16 位。该芯片还包含片上滤波、可编程控制的增益和衰减调节。

AD50 的主要特点如下：

① 单一的 5V 电源供电或 5V 模拟、3.3V 数字双电源供电。

② 内含 16 位精度的 ADC 和 DAC。

③ 器件中的 ADC 为 64 倍过采样，DAC 为 256 倍过采样(内部)。

④ ADC 和 DAC 的 SNR(信噪比)能达到 89dB。

⑤ 可编程的 ADC 和 DAC 的采样速率，最大可达 22.05kHz。

⑥ 可编程的输入输出增益。

⑦ 支持级联方式工作。

⑧ 数据的动态范围能达到 88dB。

AD50 的结构如图 12.9 所示，左边为模拟信号部分，右边为数字部分。模拟输入可以来自三种信号源：IN、AUX 和经模拟环路馈入的模拟输出信号，经复用器后，仅有一路进入 A/D 处理获得数字信号并进入缓冲，然后从 DOUT 以串行方式送出；数字输入有两个信号源：DIN 和经数字环路馈入的数字输出信号，输入信号先存放在缓冲中，然后经 D/A 处理，恢复出模拟脉冲信号，通过低通滤波器，获得最后的模拟输出信号。

ADC 通道输出的数字信号从 DOUT 送出，DAC 通道的数字输入信号从 DIN 送入。数据传送的同步时钟取自 SCLK 脚，帧同步信号取自 FS 脚。串行通信的工作方式有两种：当传送 ADC 和 DAC 数据时，采用第一串行通信方式；当要读写控制寄存器时，采用第二串行通信方式。

第一串行通信是专门用来传送 ADC 和 DAC 数据的，其数据字由寄存器 1 和 2 控制。当 D/A 数据字宽为 15 位时，最后一位是控制位，用于切换到第二串行通信方式。当 D/A 数据字宽为 16 位时，所有的 16 位都是数据，如果要切换通信方式需要由硬件完成。第二串行通信是用于设置 AD50 的。

由于 AD50 与 DSP 的 McBSP 相连，所以使用 AD50 的第一步是设置 McBSP。McBSP 的设置主要包括对接收和发送的数据格式和引脚信号的设置等。C5409 片内包括三个高速

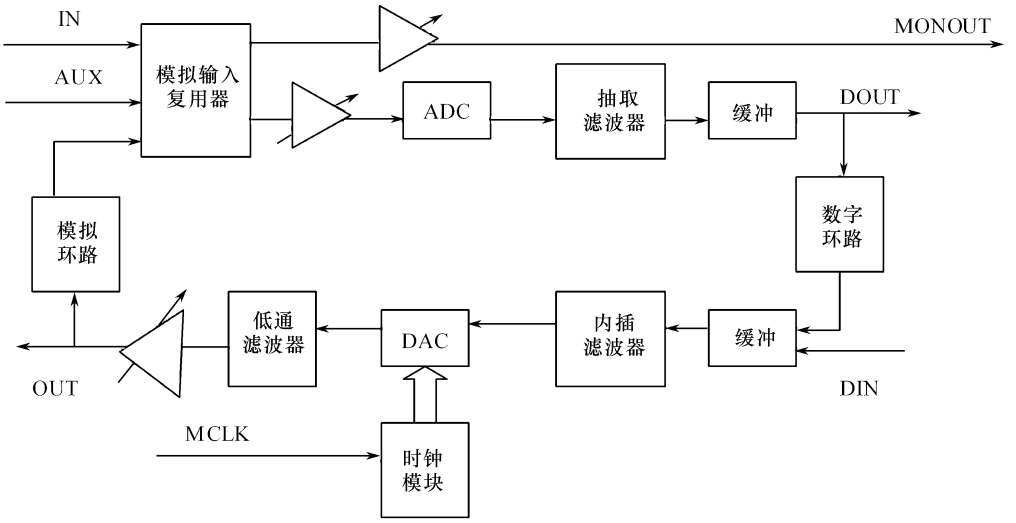


图 12.9　TLC320AD50 的结构

McBSP(多通道缓冲串口),由数据线 D(R/X),帧同步线 FS(R/X)和移位时钟线 CLK(R/X)等组成。其中 DX 完成数据的发送,DR 完成数据的接收。在时钟 CLKX、CLKR 和帧同步信号 FSR、FSX 的控制下,McBSP 通过 DX 和 DR 实现 DSP 与外部设备的通信和数据交换。本系统使用的是 McBSP0。AD50 的 DOUT 引脚与 McBSP0 的 DR 引脚相接,送入转换后的数字信号。McBSP0 的 DX 引脚与 AD50 的 DIN 引脚相接,送出要进行转换的数字信号。

3. 存储器模块

TMS320C5409 DSP 总共有 192 千字的存储器空间。这些空间可分为 3 个专门的存储空间:64 千字的程序空间、64 千字数据空间和 64 千字的 I/O 空间。程序空间一般用来存放程序代码和一些常用的系数表格;数据空间用于存放程序处理时的数据和结果;I/O 空间可以映射为外部设备,也可以用于扩展外部数据存储器。每个空间中都可以包含不同的存储器类型,如 RAM、ROM、EPROM 或存储器映像外设等,这些部分可以来自于片内,也可以来自于片外。C5409 存储器的配置受处理器模式状态寄存器(PMST)的三个控制位 MP/$\overline{\text{MC}}$、OVLY 和 DROM 的影响。具体影响如下:

(1) MP/$\overline{\text{MC}}$位:微处理器/微型计算机工作方式位

当 MP/$\overline{\text{MC}}$=0 时,允许片内 ROM 配置到程序存储空间;

当 MP/$\overline{\text{MC}}$=1 时,禁止片内 ROM 配置到程序存储空间。

(2) OVLY:RAM 重叠位

当 OVLY=1 时,片内 RAM 配置到程序和数据存储空间;

当 OVLY=0 时,片内 RAM 仅配置数据存储空间。

(3) DROM 位:数据 ROM 位

当 DROM=1 时,片内 ROM 配置到程序和数据存储空间;

当 DROM=0 时,禁止 ROM 配置到数据存储空间。

图 12.10 是 TMS320C5409 芯片数据和程序存储空间的配置图,从图中可以看到上述三个控制位与存储器配置的关系。

TMS320C5409 的内部仅有 32 千字的 RAM 和 16 千字的 ROM，远不能满足需求，因此系统扩展了两片 64 千字的 SRAM 用来存放语音数据，并且还扩展了一片 512 千字的 Flash 用来存放程序。其中 Flash 使用的是 AT49LV8192A-90TC，SRAM 使用的是 T71V016SA10Y。

图 12.10　C5409 的存储空间的映射

C5409 可以将程序存储空间采用分页扩展方法扩展到 8 兆字。扩展程序存储空间的配置如图 12.11 所示。

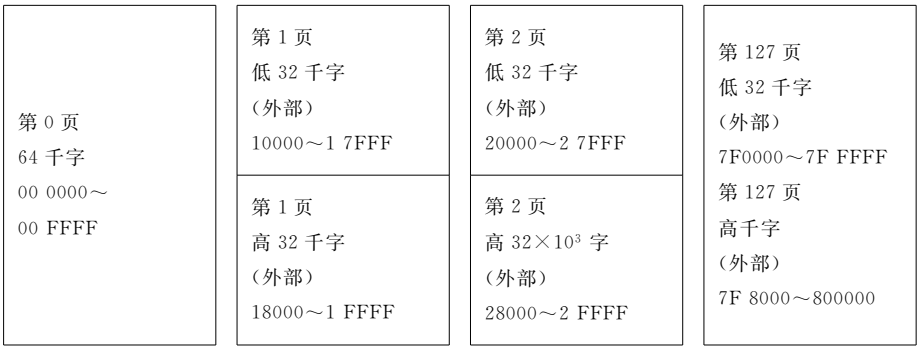


图 12.11　扩展程序空间的映射

4. 异步通信串口模块

异步通信串口(UART)模块用来实现 DSP 与计算机的通信。该模块主要由 TL16C550

和 MAX3238 组成。其中 MAX3238 完成电平转换，TL16C550 完成数据传送的并/串转换以及串行传输的波特率设定等功能。对 DSP 而言，能够访问的只有 TL16C550，这部分主要介绍 DSP 编程访问 TL16C550 实现 UART 的具体过程。

TI 公司的 TL16C550 是一个 RS232 串行口协议转换芯片，对主机而言它是并行口访问的，所有收发数据都存放在各自的 FIFO 中，对另一端设备而言它是串行的，收发数据都符合 RS232 协议。其结构如图 12.12 所示。左边是数字接口部分，右边是 RS232 接口部分。

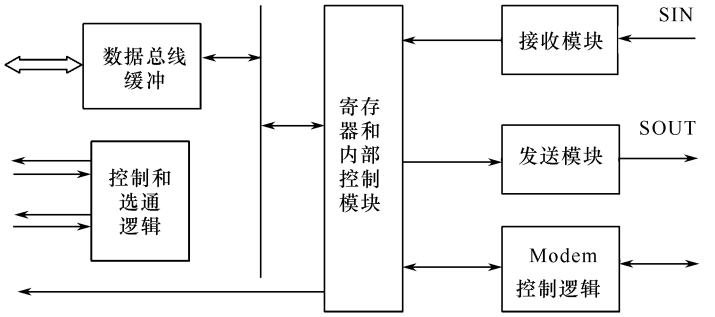


图 12.12　TL16C550 的结构图

主机通过并行方式访问 TL16C550 的寄存器，寄存器的设定将控制其内部的控制逻辑模块，实现对其工作方式的设置（如波特率、校验位等），同时通过访问寄存器也可以实现对数据的操作（读取和写入数据）。RS232 数据的接口，可大致分为三个部分：接收模块、发送模块和 Modem 控制逻辑。接收模块将从 SIN 引脚输入的串行数据按照规定的格式，取出其数据部分并作校验，数据部分被送入接收寄存器或接收 FIFO 中，校验结果反应在状态位上。发送模块将发送寄存器或发送 FIFO 中的数据按照规定的格式加入起始位、停止位和校验位，并以 RS232 的串行方式发送至 SOUT 引脚。Modem 控制逻辑通过接收和发送引脚信号，实现对收发操作的控制。

从 SIN 进入的数据首先进入接收移位寄存器（RSR），一个字符接收完成后，数据移入 RBR。RBR 实际是一个 16 字节的 FIFO，当中断设置时，TL16C550 会根据 FIFO 中接收数据的数目产生中断。如果主机设备从 RBR 中读取数据后，中断会自动清除。

发送操作与接收操作相反，主机数据写入 THR，THR 是一个 16 字节的 FIFO。然后数据移入传输移位寄存器（TSR），最后送至 SOUT。当中断设置时，TL16C550 会根据 FIFO 中发送数据的数目产生中断。主机设备可根据中断来决定是否继续发送数据。

当使能 FIFO 后，TL16C550 工作方式有两种：中断方式和查询方式。

TL16C550 一般工作在中断模式下，通过向 DSP 发送中断信号，通知 DSP 通过并口向 TL16C550 发送或从 TL16C550 接收 8bit 的数据。TL16C550 按设定的波特率与计算机进行通信。

TL16C550 进行串行通信的波特率设置是通过除数寄存器（DLM 和 DLL）来实现的。实际的波特率为输入时钟信号进行分频后得到，计算公式如下：

$$\text{BaudRate}=\frac{\text{CLKIN}}{\text{divisor}} \tag{12.1}$$

其中，divisor 为除数寄存器的数值。本系统提供给 TL16C550 的时钟为 1.384MHz，波特率设置为 9600Baud。

5. **JTAG 模块**

JTAG 模块是为外部设备提供控制 DSP 工作的接口，它的结构如图 12.13 所示。它包括两个主要部分：内嵌的 JTAG TBC 和外接 JTAG 接口。TBC 是 SN74ACT8990，它提供了并行方式控制 JTAG 的接口，主机可以通过 PPC 访问 TBC，完成主机对 DSP 的调试。对于主机 PC 软件来说，TBC 是通过 IEEE-1284 并行口和 PPC34C60 访问的 24 个存储器映射的存储器。外接 JTAG 接口使用户可通过外部仿真器 XDS510、XDS510PP 直接调试 DSP。

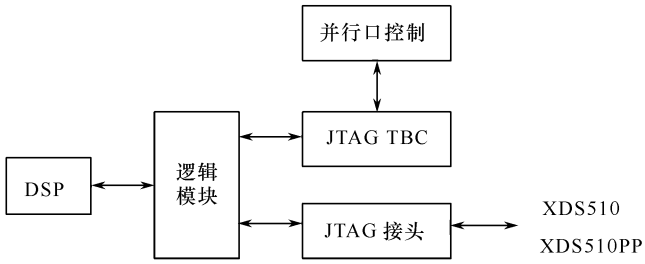


图 12.13　JTAG 模块

6. **电源模块**

DSP 板上所有的电源都是通过外部馈入的 5V 直流电源经过变换后获得的。该电源送到 DSP 板上后先隔离，然后分别形成 5V 的数字电源和 5V 的模拟电源。5V 的数字电源经 LM4040DCIM3-4.1 分压生成 4.1V 的数字电源。其他电压是通过使用 TPS70351 电压调节器获得的。TPS70351 的输入是 5V 的直流电源，输出为 3.3V 和 1.8V。其中 3.3V 提供板上的数字电源，1.8V 提供 DSP 的工作电压。

12.5.2　软件设计

基于 C54x DSP 的软件设计可用 C 语言编程及混合编程，其软件设计和代码生成的步骤如下：

- 用 C 语言或汇编语言编写程序，后缀分别为.c 或.asm。
- 用 C 编译器把.c 文件编译成.asm 文件，生成.obj 文件，或者用汇编器把.asm 生成.obj 文件。
- 用链接器根据链接命令(.cmd)文件将多个.obj 文件、库文件链接起来，并分配各程序段、数据段的地址，生成的.out 文件可供模拟/仿真。

设计者可以根据设计要求采用 C、汇编或两种语言混合编程。

1. **C54x DSP 的 C 语言编程**

目前流行的 DSP 设计方法是用 C 语言，其原因是由于 C 语言易学易懂，可移植性强。各种 DSP 都支持 C 语言设计，DSP 厂商提供了丰富的 C 库函数，设计者可以按照标准 C 语言语法编写处理程序，用标准 C 语言编写处理程序可以不必了解信号处理是如何由 DSP 的硬件和指令完成的，这大大方便了 DSP 的开发。C 语言的缺点是编译效率低，代码长，造成处理性能下降，存储器需求上升。然而 DSP 速度的提高抵消了 C 编译器的不足。DSP 片内 RAM 容量不断加大、片外 RAM 的体积和成本迅速下降，使得存储器资源不再约束 DSP 设计者。在厂商提供的算法库中，对常用的算法提供了手工汇编和优化。这些都有助于设计者采用 C 语言

编写软件程序，使其能专注于 DSP 软件编程，这也符合当前电子设计软件化的发展要求。因此，未来 DSP 的理想设计模式是：设计人员编写通用的 C 程序，当 DSP 或 DSP 电路板升级时，只要将程序在新的 DSP 开发环境下重新编译和链接，就能生成与实际电路匹配的可执行代码。这样的开发过程是十分快捷的，也节省了大量的人力资源、软件开发与维护费用。因此，各大 DSP 生产商都相继推出了 C 语言编译器，如 TI 公司的 CCS 集成开发环境能够编译 C 和 C++语言。

C 语言程序经 C 编译器编译后，自动输出如下代码和数据段：

.text 段：包含可执行代码和编译器产生的常数。

.cinit 段：包含用来初始化变量和常数的表。

.const 段：包含字符串和用 const 定义的数据常数。

.bss 段：存放静态和全局变量，在加载过程中，加载程序会从 .cinit 段中复制数据用来初始化这些静态和全局变量。

.stack 段：系统堆栈存储空间。用于变量传递及分配局部变量。

.sysmem 段：动态分配存储器。如果 C/C++程序中没有动态分配程序空间，此段就不会产生。

一般.text、.cinit 和.const 段链入到系统的 ROM 或 RAM 中，.bss、.stack 和.sysmem 段链入到系统的 RAM 中。

2. C 语言和汇编语言的混合编程方法

① 独立编写 C 程序和汇编程序，分开编译或汇编以形成各自的目标代码模块，然后用链接器将 C 模块和汇编模块链接起来。例如，主程序用 C 语言编写，硬件初始化用汇编语言编写。如果要在 C 程序中访问汇编程序的变量，将汇编语言程序在 .bss 块中定义的变量或函数名前加一下画线，将变量说明为外部变量，同时在 C 程序中也将变量说明为外部变量。如汇编中的变量不在.bss 段中定义，例如利用.usect 定义的段中，则在 C 程序中需利用一个指针来访问此变量。

② 在 C 语言程序的相应位置直接嵌入汇编语句，这是一种 C 语言和汇编之间比较直接的接口方法。采用这种方法一方面可以在 C 语言中实现用 C 语言不好实现的一些硬件控制功能，如中断使能或禁止等；另一方面，也可以用这种方法在 C 程序中的关键部分用汇编语句代替 C 语句以优化程序。在 C 程序中嵌入汇编指令时，一定要注意不要破坏 C 程序的环境。

③ 对 C 程序进行编译生成相应的汇编程序，然后对汇编程序进行手工优化和修改。这种方法可以通过查看交叉列表的汇编程序，可以对某些不是很优却是比较关键的汇编语句进行修改。

系统采用了前两种方法进行混合编程。

3. 软件设计

图 12.14 为系统主程序流程图。为了便于程序的设计与调用，采用了模块化的程序设计方法。整个编程思想是用 C 语言构建整个系统的框架程序，而所用到的算法都用子程序模块来实现，对于频繁被调用的子程序，如 FIR 滤波，采用汇编语言实现。为了充分利用 DSP 独特的硬件结构，中断服务程序及对硬件的初始化采用汇编语言编写，其他的用 C 语言来实现。软件算法由 C 语言和汇编语言混合编程，采用混合编程方法，即可缩短软件开发的周期，提高程序的可读性和可移植性，又能满足系统实时运算的要求。

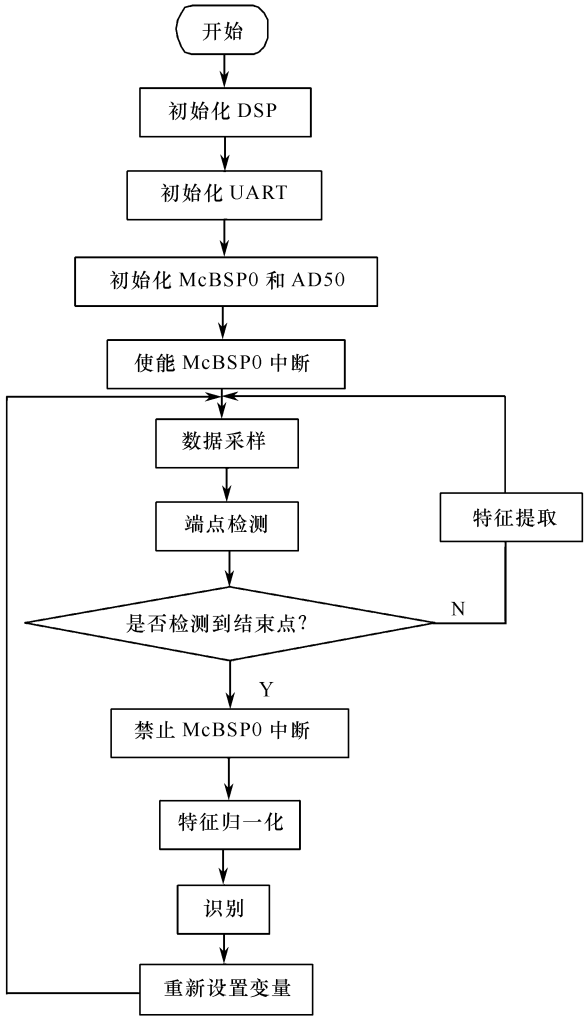


图 12.14　系统主程序流程图

(1) DSP 的初始化

初始化 DSP 是完成对 DSP 堆栈,CPU 时钟以及其他各个工作寄存器的初始值设置,以满足系统工作要求。初始化 SWWSR、BSCR、ST0、ST1、PMST 等寄存器,设置中断屏蔽寄存器 IMR,屏蔽所有的中断,并置 IFR＝0xFFFF。DSP 的外部时钟为 20MHz,采用 PLL 方式,并将其倍频系数设置为 5,让 DSP 的工作频率为 100MHz。

(2) UART 的初始化

初始化 UART 程序是完成对异步通信串口波特率、数据格式及工作方式的设置。UART 芯片 TLC16C550C 的寄存器都被映射到 C5409 的 I/O 空间,且起始地址为 0x0000,所以应该按照 I/O 端口的方式访问寄存器。设置 UART 的时钟为 1.8432MHz,通过设置除数寄存器使 UART 的波特率为 9600Baud。采用无奇偶校验,字符长度为 8,停止位为 1 的数据格式,数据的接收和发送均采用查询方式。

(3) McBSP0 和 AD50 的初始化

初始化 McBSP0 和 AD50 程序是用来设置 McBSP0 和 AD50 的工作状态。

C5409 的 McBSP0 由 SPCR、RCR、XCR、SRGR 等寄存器控制。初始化串口的步骤如下:

① 复位 McBSP0 并设置控制寄存器的帧同步信号和串口时钟均为 External，设置接收中断信号由帧同步信号产生，用中断方式来向 McBSP0 发送数据，使能串口的中断。

② 设置 McBSP0 的引脚控制寄存器，是串口的所有引脚都工作在串行口方式，而不是通用的 I/O 方式。

③ 设置 McBSP0 的发送和接收控制寄存器，使接收到的每一帧包含一个字。

④ 使能全部中断，并使串口脱离复位状态。

使 McBSP0 运行在从动模式下，接收和发送时钟以及帧同步信号都由 AD50 的时钟和帧同步信号驱动，每帧 16bit，接收和发送数据都没有延时。

初始化 AD50 之前，首先置 AD50 的复位信号为 0，用于复位 AD50，使得 AD50 设置为默认配置状态。在 C5409 和 McBSP0 初始化完成后，将 AD50 的复位信号置 1，使 AD50 脱离复位状态。通过设置 AD50 的寄存器，使 AD50 的采样率为 8kHz，采用 15bit 的模式，输入输出增益均为 0。

McBSP0 采用中断的方式接收采样后的数据。初始化操作完成后，使能 McBSP0 的接收中断。DSP 把接收到的数据存放到数据空间的缓冲区中，大小为 100×16bit。此缓冲区为循环缓冲区，以便用一个有限容量的数据区来存储数量极大的语音数据，已处理完提取出了语音特征参数的一个时间段的语音数据可以依次抛弃，让出存储空间来存储新的数据。

将 McBSP0 接收到的语音信号利用短时能量和过零率相结合的方法进行端点检测。判断完语音起止点后，进行特征提取。将提取到的特征送到 RBF 神经网络识别。最后在 PC 上实时显示识别结果。

12.5.3 独立系统形成

软硬件调试成功后，首次通过仿真器将用户 BOOT 程序写入 Flash，然后再通过仿真器将整个系统的软件写入 Flash。两个软件写入 Flash 后，系统就可以成为独立(即不再和仿真器相连接)运行的 DSP 系统。

本章介绍的语音识别系统在正确写入 Flash 后，用 9 针连接线使系统 UART 与计算机相连(计算机上事先应装有相应的 UART 接口调试和显示工具)，加电启动后，系统就进入语音识别状态，试验结果表明中等词汇量识别率在 85％以上。

附录 A　专业术语缩写英汉对照表

缩写	英文名称	中文名称
	A	
ACELP	Algebraic Code Excited Linear Prediction	代数码激励线性预测
ADM	Adaptive Delta Modulation	自适应增量调制
ADPCM	Adaptive Differential Pulse Code Modulation	自适应差分脉冲编码调制
AMDF	Average Magnitude Difference Function	平均幅度差函数
AMI	American Megatrends Incorporation	美国趋势公司
AMR-NB	Adaptive Multi Rate-Narrowband	自适应多速率窄带
AMR-WB	Adaptive Multi Rate-Wideband	自适应多速率宽带
ANN	Artificial Neural Network	人工神经网络
APC	Adaptive Predictive Coding	自适应预测编码
APCM	Adaptive Pulse Code Modulation	自适应脉冲编码调制
API	Application Programming Interface	应用程序接口
APVQ	Adaptive Predictive Vector Quantization	自适应预测矢量量化
AR	Auto Regressive	自回归
ARMA	Auto Regressive Moving Average	自回归滑动平均
	B	
BIOS	Basic Input Output System	基本输入输出系统
BP	Back-Propagation	反向传播
BSD	Bark Spectral Distortion	巴克谱失真
	C	
CCITT	International Telegraph and Telephone Consultative Committee	国际电报电话咨询委员会
CCS IDE	Code Composer Studio Integrated Development Environment	TI 公司用于 DSP 电路调试及程序调试的一种集成开发环境
CDMA	Code Division Multiple Access	码分多址
CELP	Code Excited Linear Prediction	码激励线性预测编码
CNG	Comfort Noise Generator	舒适噪声生成器

CS-ACELP	Conjugate Structure-Algebraic Code Excited Linear Prediction	共轭结构代数码激励线性预测编码
CTS	Concept-To-Speech	从概念到语音
CVSD	Continuously Variable Slop Delta Modulation	连续可变斜率增量调制
	D	
DAM	Diagnostic Acceptability Measure	判断满意度测量
DCT	Discrete Cosine Transform	离散余弦变换
DDBHMM	Duration Distribution Based Hidden Markov Model	基于段长分布的隐含马尔可夫模型
DDN	Digital Data Network	数字数据网
DEC	Digital Equipment Corporation	数字设备公司
DFT	Discrete Fourier Transform	离散傅里叶变换
DM	Delta Modulation	增量调制
DMA	Direct Memory Access	直接存储器访问
DP	Dynamic Programming	动态规划
DPCM	Differential Pulse Code Modulation	差分脉冲编码调制
DRT	Diagnostic Rhyme Test	判断韵字测试
DSK	DSP Starter Kit	DSP 入门套件
DSP	Digital Signal Processing	数字信号处理
	Digital Signal Processor	数字信号处理器
DTW	Dynamic Time Warping	动态时间弯折(规整)
DTX	Discontinuous Transmission	不连续传输
	E	
ECU	Error Concealment Units	差错隐藏单元
EVM	Evaluation Module	评价模块
EVRC	Enhanced Variable Rate Codec	增强型可变速率编码器
	F	
FDM	Frequency Division Multiplexing	频分复用
FDMA	Frequency Division Multiple Access	频分多址
FD-PSOLA	Frequency Domain- Pitch Synchronous Overlap Add	频域基音同步叠加
FFT	Fast Fourier Transform	快速傅里叶变换
FIR	Finite duration Impulse Response	有限冲激响应

FLOPS	Floating point number Operations Per Second	每秒浮点运算次数
FSK	Frequency Shift Keying	频移键控信号
	G	
3GPP	3rd Generation Partnership Project	第三代合作伙伴计划
GPS	Global Position System	全球定位系统
GSM	Global System for Mobile communication	全球移动通信系统
	H	
HMM	Hidden Markov Model	隐马尔可夫模型
HPI	Host Port Interface	主机接口
	I	
IDFT	Inverse Discrete Fourier Transform	离散傅里叶逆变换
IFFT	Inverse Fast Fourier Transform	快速傅里叶逆变换
IP	Internet Protocol	互联网协议
ISDN	Integrated Services Digital Network	综合服务数字网
ISF	Immittance Spectrum Frequency	导抗谱频率
ISP	Immittance Spectral Pair	导抗谱对
ITS	Intention-To-Speech	从意向到语音
ITU-T	International Telecommunication Union-Telecommunication Sector	国际电信联盟-电信标准部
	K	
KLT	karhunen-loéve transform	卡亨南-洛维变换
	L	
LD-CELP	Low-Delay Code-Excited Linear Prediction	低延时码激励线性预测
LAR	Log Area Ratios	对数面积比
LPC	Linear Predictive Coding	线性预测编码
LPCC	Linear Predictive Cepstral Coefficient	线性预测倒谱系数
LPC-PSO-LA	Linear Predictive Coding- Pitch Synchronous Overlap Add	线性预测基音同步叠加
LSF	Linear Spectrum Frequency	线谱频率
LSP	Linear Spectral Pair	线谱对
LTP	Long-Term Prediction	长时预测
LVQ	Learning Vector Quantization	学习矢量量化
	M	
MA	Moving Average	滑动平均
McBSP	Multichannel Buffered Serial Port	多通道缓冲串口
MELPC	Mixed Excitation Linear Prediction Coding	混合激励线性预测编码

MFCC	Mel-Frequency Cepstrum Coefficient	Mel 频率倒谱系数
MIPS	Million Instructions Per Second	百万条指令/秒
MMSE	Minimum Mean Square Error	最小均方误差
MPE-LPC	Multi Pulse Excited-LPC	多脉冲激励线性预测编码
MOPS	Million Operations Per Second	百万次操作/秒
MOS	Mean Opinion Score	平均意见得分
MOS-LQO	Mean Opinion Score-Listening Quality Objective	平均意见得分-客观听觉质量

N

NNR	Nearest Neighbor Rule	最近邻法

O

OMAP	Open Multimedia Application Platform	开放式多媒体应用平台

P

PARCOR	Partial Correlation Coefficient	部分相关系数
PCM	Pulse Code Modulation	脉冲编码调制
PDA	Personal Digital Assistant	个人数字助理
PESQ	Perceptual Evaluation of Speech Quality	感知语音质量评价
PLP	Perceptual Linear Prediction	感知线性预测
PSELP	Pitch Synchronous Excited Linear Prediction	基音同步激励线性预测
PSOLA	Pitch Synchronous Overlap Add	基音同步叠加

Q

QCELP	Qualcomm Code Excited Linear Prediction	Qualcomm 公司的码激励线性预测

R

RAM	Random Access Memory	随机存储器
RBF	Radial Basis Function	径向基函数
RC	Reflection Coefficient	反射系数
RDA	Rate Decision Algorithm	速率判决算法
RELP	Residual Excited Linear Prediction	残差激励线性预测
RMS	Root Mean Square	均方根
RNN	Recurrent Neural Network	递归神经网络
ROM	Read Only Memory	只读存储器
RPE-LPC	Regular Pulse Excited-LPC	规则脉冲激励线性预测编码
RTDX	Real-Time Data Exchange	实时数据交换

S

SB-ADPCM	Sub Band- ADPCM	子带自适应差分脉码调制

SMV	Selectable Mode Vocoder	可选模式声码器
SNR	Signal Noise Ratio	信噪比
SONN	Self-Organizing Neural Network	自组织神经网络
STC	Sinusoidal Transform Coding	正弦变换编码
	T	
TC	Transform Code	变换编码
TCP	Transmission Control Protocol	传输控制协议
TDM	Time Division Multiplexing	时分复用
TDMA	Time Division Multiple Access	时分多址
TDNN	Time Delay Neural Network	时间延迟神经网络
TD-PSOLA	Time Domain-PSOLA	时域基音同步叠加
TFI	Time Frequency Interpolation	时频插值
TI	Texas Instruments	美国得克萨斯仪器公司
TTS	Text To Speech	文本到语音
	V	
VAD	Voice Activity Detector	语音激活检测器
VMR-WB	Variable-Rate Multimode Wideband	变速率多模式宽带
VoIP	Voice over Internet Protocol	网络电话
VQ	Vector Quantization	矢量量化
VSELP	Vector Sum Excited Linear Predictive Coding	矢量和激励线性预测编码
	W	
WCDMA	Wideband Code Division Multiple Access	宽带码分多址
	Z	
ZCPA	Zero-Crossing with Peak-Amplitudes	过零峰值幅度

参 考 文 献

[1] 赵力. 语音信号处理. 北京:机械工业出版社,2003.

[2] 蔡莲红,黄德智,蔡锐. 语音技术基础与应用. 北京:清华大学出版社,2003.

[3] 胡航. 语音信号处理. 哈尔滨:哈尔滨工业大学出版社,2000.

[4] 杨行峻,迟惠生等. 语音信号处理. 北京:电子工业出版社,1995.

[5] 蔡莲红,黄德智,蔡锐. 现代语音技术基础与应用. 北京:清华大学出版社,2003.

[6] Sanjit K. Mitra 著,孙洪,余翔宇等译. 数字信号处理(基于计算机的方法). 北京:电子工业出版社,2005.

[7] 姚天任. 数字语音处理. 武汉:华中科技大学出版社,2003.

[8] 易克初,田斌,付强. 语音信号处理. 北京:国防工业出版社,2000.

[9] 王炳锡,屈单,彭煊. 实用语音识别基础. 北京:机械工业出版社,2005.

[10] 韩纪庆,张磊,郑铁然. 语音信号处理. 北京:清华大学出版社,2004.

[11] 王让定,柴佩琪. 语音倒谱特征的研究. 计算机工程,2003,29(13):31～33.

[12] [美]L. R. 拉宾纳,R. W. 谢佛. 语音信号数字处理. 北京:科学出版社,1983.

[13] Oppenheim AV, Schafer RW. Discrete-time Signal Processing, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[14] 张刚,张雪英,马建芬. 语音处理与编码. 北京:兵器工业出版社,2000.

[15] 孟飚. 8kbit/s CS-ACELP 语音编码算法的研究与实现. 太原:太原理工大学,2003.

[16] 白国栋. 自适应多速率宽带语音编码算法的仿真实现及研究. 太原:太原理工大学,2008.

[17] 王炳锡,王洪. 变速率语音编码. 西安:西安电子科技大学出版社,2004.

[18] 鲍长春. 低比特率数字语音编码基础. 北京:北京工业大学出版社,2001.

[19] 温斌等. 中低速率语音编码技术的发展及应用. 电信科学,1996,(10):35～38.

[20] [美]J. D. 马卡尔等编著,娄乃英译. 语音信号线性预测. 北京:中国铁道出版社,1987.

[21] Recommendation G729,Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP). Geneva, Switzerland: ITU-T, March 1996.

[22] Recommendation P-862,Perceptual Evaluation of Speech Quality (PESQ)-An Objective Method for End-to-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs. Geneva,Switzerland: ITU-T, 2001.

[23] Recommendation G. 721,A 32kbit/s Adaptive Differential Pulse-Code-Modulation(ADPCM). Red Books, CCITT, 1984.

[24] Oded Ghitza. Auditory Models and Human Performance in Tasks Related to Speech Coding and Speech Recognition, IEEE Transactions on Speech and Audio Processing, 1994, 2(1): 13～131.

[25] Recommendation G. 722. 2, Wideband Coding of Speech at Around 16kbit/s Using Adaptive Multi-Rate Wideband (AMR-WB). ITU-T, 2003.

[26] TS 26. 190. Adaptive Multi-Rate Wideband Speech Codec: Transcoding functions.

3GPP, 2001.
[27] 李娟. 基音周期检测算法研究及在语音合成中的应用. 太原:太原理工大学,2008.
[28] 柏静,韦岗. 一种基于线性预测与自相关函数法的语音基音周期检测算法. 语音技术,2005, 43(4):42～45.
[29] M. J. Ross, H. L. Shaffer, A. Cohen, etal. Average Magnitude Difference Function Pitch Extractor, IEEE Trans. on Acoustics Speech and Signal Proc, 1974, 22(5): 353～362.
[30] 鲍长春,樊昌信. 基于归一化互相关函数的基音检测算法. 通信学报,1998, 19(10): 27～31.
[31] Yu-Min Zeng, Zhen-Yang Wu, Hai-Bin Liu, Lin Wu. Modified AMDF Pitch Detection Algorithm, Proceedings of the Second International Conference on Machine Learning and Cybernetics, November 2003, 1: 470～473.
[32] 王晓亚. 倒谱在语音的基音和共振峰提取中的应用. 无线电工程,2004,34(01):57～61.
[33] 朱维彬,吕士楠. 基于语义的语音合成-语音合成技术的现状及展望. 北京理工大学学报,2007,27(5):408～412.
[34] 陶建华,蔡莲红. 计算机语音合成的关键技术及展望. 计算机世界,2000,(3):20.
[35] 张后旗,俞振利,张礼和. 基于 TD-PSOLA 算法的汉语普通话韵律合成. 科技通报,2002,18(1):6～9.
[36] 刘建,郑方,邓箐,吴文虎. 基于混合幅度差函数的基音提取算法. 电子学报,2006,34(10):1925～1928.
[37] 方青,国辛纯,洪锐. TD-PSOLA 算法对基音频率和时长的控制. 电子测量技术,2006,29(6):175～176.
[38] 俞铁城. 语音识别的发展现状. 通信世界,2005,(2):56～57.
[39] 拉宾纳. 语音识别的基本原理. 北京:清华大学出版社,2002.
[40] 郑方. 非特定人连续数字识别方法与汉语语音数据库的研究. 北京:清华大学,1992.
[41] 白静,张雪英,侯雪梅. 基于 RBF 神经网络的抗噪语音识别. 计算机工程与应用,2007,43(22):28～30.
[42] Doh-Suk Kim, Soo-Yong Lee, Rhee M. Kil. Auditory Processing of Speech Signals for Robust Speech Recognition in Real World Noisy Environments, IEEE Transactions on Speech and Audio Processing, 1999, 7(1): 55～69.
[43] Rabiner L R. A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition, Proc. of the IEEE, 77(2), 1989: 257～286.
[44] Bojana Gajic. Robust Speech Recognition Using Feature Based on Zero Crossing with Peak Amplitudes, ICASSP, 2003: 64～67.
[45] 赵姝彦. 基于 ZCPA 和 DHMM 的孤立词语音识别系统. 太原:太原理工大学,2005.
[46] 焦志平. 改进的 ZCPA 语音识别特征提取算法研究. 太原:太原理工大学,2005.
[47] 梁五洲. 抗噪语音识别特征提取算法的研究. 太原:太原理工大学,2006.
[48] 杨行峻,郑君里. 人工神经网络与盲信号处理. 北京:清华大学出版社,2003.
[49] 赵明忠,顾斌等. DSP 应用技术. 西安:西安电子科技大学出版社,2004.
[50] 周霖. DSP 系统设计与实现. 北京:国防工业出版社,2003.
[51] 张勇,曾炽祥,周好斌,陈滨. TMS320C5000 系列 DSP 汇编语言程序设计. 西安:西安电

子科技大学出版社，2004.
[52] 徐盛，胡剑凌. 数字信号处理器. 上海：上海交通大学出版社，2003.
[53] 梁芳泉. 抗噪语音识别算法的 DSP 实现. 太原：太原理工大学，2006.
[54] Texas Instruments, TMS320VC54x DSP Library Programmer's Reference, 2004.
[55] Texas Instruments, TMS320VC54x DSP Reference Set, Vol. 1: CPU and Peripherals, 1999.
[56] Texas Instruments, TMS320VC54x DSP Reference Set, Vol. 2: Mnemonic Instruction Set, 1999.
[57] Texas Instruments, TMS320VC5409 Fixed-Point Digital Signal Proeessor Data Manual, 2004.
[58] Texas Instruments, TMS320VC54x DSP Reference Set, Vol. 4: Applications Guide, 1999.
[59] Texas Instruments, TMS320VC54x Assembly Language Tools User's Guide, 1998.
[60] Ye Li, Hui-juan Cui, Kun Tang. Speech Enhancement Algorithm Based on Spectral Subtraction, Journal of Tsinghua University, October 2006, 46(10): 1685～1687.
[61] Martin Rainer. Speech Enhancement Based on Minimum Mean-Square Error Estimation and Super Gaussian Priors, IEEE Transactions on Speech and Audio Processing, September 2005, 13(5): 845～856.
[62] 杨海. 感知语音质量评价 PESQ 及其在通信系统中的应用. 江西通信科技，2004，(2)：46～47.
[63] 陈照平. 基于短时谱估计的语音增强方法研究. 太原：太原理工大学，2008.
[64] Cohen, I. Noise Spectrum Estimation in Adverse Environments: Improved Minima Controlled Recursive Averaging, IEEE Trans, Speech Audio Process, 2003, 11 (5): 466～475.
[65] Sundarrajan Rangachari, Philipos C. Loizou. A Noise-Estimation Algorithm for Highly Non-Stationary Environments, Speech Communication, 2006, 48(2): 220～231.
[66] Yariv Ephraim, David Malah. Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator, IEEE Transactions on Acoustics, Speech and Signal Processing, December 1984, 32(6): 1109～1121.
[67] 赵晓群. 数字语音编码. 北京：机械工业出版社，2007.
[68] 李昌立，吴善培. 数字语音-语音编码实用教程. 北京：人民邮电出版社，2004.
[69] 胡征，杨有为. 矢量量化原理与应用. 西安：西安电子科技大学出版社，1988.
[70] 李凤莲，张雪英. ISP 与 LSP 的特性比较. 太原理工大学学报，2008，(39)：581-584.
[71] Stephen So, Kuldip K. Paliwal. A Comparative Study of LPC Parameter Representations and Quantisation Schemes for wide-band Speech Coding. Digital Signal Processing, 2007, 17(1), 114～137.
[72] Yuval Bistritz, Shlomo Peller. Immittance Spectral Pairs (ISP) for Speech Encoding. IEEE Int Conf Acoust, 1993 (2): 9～12.

电子信息与电气学科规划教材·电子信息科学与工程类专业

数字语音处理及MATLAB仿真

本书系统地阐述了语音信号处理的原理、方法、技术和应用，同时给出了部分内容对应的MATLAB仿真源程序。全书共12章，第1章至第7章是基本理论部分，包括语音信号的数字模型、语音信号的短时时域分析和频域分析、语音信号的同态处理、语音信号线性预测分析和矢量量化；第8章至第12章是应用部分，包括语音编码、语音合成、语音识别、语音增强和语音处理的实时实现。本书内容全面，重点突出，原理阐述深入浅出，注重理论与实际应用的结合，可读性强。

本书可作为高等院校通信工程、电子信息工程、自动控制、计算机技术与应用等专业高年级本科生相关课程的教材，也可供从事语音信号处理研究的研究生和科研人员参考。

策划编辑：凌　毅
责任编辑：李秦华
责任美编：孙焱津

ISBN 978-7-121-11323-9
9 787121 113239 >
定价：　　元